

**NASA TECHNICAL
MEMORANDUM**



NASA TM X-2434
C.1

NASA TM X-2434

**LOAN COPY: RE
AFWL (DO
KIRTLAND AF**

0151983



TECH LIBRARY KAFB, NM

**COMPUTER PROGRAM FOR DESIGN
OF TWO-DIMENSIONAL SUPERSONIC
TURBINE ROTOR BLADES
WITH BOUNDARY-LAYER CORRECTION**

by Louis J. Goldman and Vincent J. Scullin

Lewis Research Center

Cleveland, Ohio 44135



0151983

1. Report No. NASA TM X-2434		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle COMPUTER PROGRAM FOR DESIGN OF TWO-DIMENSIONAL SUPERSONIC TURBINE ROTOR BLADES WITH BOUNDARY-LAYER CORRECTION				5. Report Date December 1971	
				6. Performing Organization Code	
7. Author(s) Louis J. Goldman and Vincent J. Scullin				8. Performing Organization Report No. E-6494	
9. Performing Organization Name and Address Lewis Research Center National Aeronautics and Space Administration Cleveland, Ohio 44135				10. Work Unit No. 113-34	
				11. Contract or Grant No.	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546				13. Type of Report and Period Covered Technical Memorandum	
				14. Sponsoring Agency Code	
15. Supplementary Notes					
16. Abstract <p>A FORTRAN IV computer program for the design of two-dimensional supersonic rotor blade section corrected for boundary-layer displacement thickness is presented. The ideal rotor is designed by the method of characteristics to produce vortex flow within the blade passage. The boundary-layer parameters are calculated by Cohen and Reshotoko's method for laminar flow and Sasman and Cresci's method for turbulent flow. The program input consists essentially of the blade surface Mach number distribution and total flow conditions. The primary output is the corrected blade profile and the boundary-layer parameters.</p>					
17. Key Words (Suggested by Author(s)) Supersonic rotor blade design Supersonic flow				18. Distribution Statement Unclassified - unlimited	
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 58	
				22. Price* \$3.00	

COMPUTER PROGRAM FOR DESIGN OF TWO-DIMENSIONAL SUPERSONIC TURBINE ROTOR BLADES WITH BOUNDARY-LAYER CORRECTION

by Louis J. Goldman and Vincent J. Scullin

Lewis Research Center

SUMMARY

A FORTRAN IV computer program for the design of two-dimensional supersonic rotor blade sections corrected for boundary-layer displacement thickness is presented. The ideal rotor is designed by the method of characteristics to produce vortex flow within the blade passage. The boundary-layer parameters are calculated by Cohen and Reshotko's method for laminar flow and Sasman and Cresci's method for turbulent flow. The program input consists essentially of the blade surface Mach number distribution and total flow conditions. The primary output is the corrected blade profile and the boundary-layer parameters.

INTRODUCTION

Methods for the design of supersonic turbines for possible use in turbopump and open-cycle auxiliary power systems, where high-energy fluids are used and high pressure ratios are available, have recently become of interest. The method of characteristics as applied to the two-dimensional isentropic flow of a perfect gas can be used for the design of the supersonic blading. Computer programs for the isentropic design of two-dimensional supersonic nozzles and rotor blade sections have been described in references 1 and 2, respectively. A computer program for the design of two-dimensional supersonic nozzles with boundary-layer correction is described in reference 3.

This report presents a computer program for the design of supersonic rotor blades where losses are accounted for by correcting the rotor profile for boundary-layer displacement thickness. The ideal rotor blade profile is obtained by using the computer program described in reference 2. Boundary-layer parameters are calculated by using the computer program described in reference 4. The final rotor blade profile is then

obtained by adding the displacement thicknesses to the ideal blade coordinates. The program described herein is essentially a modified combination of the two programs described in references 2 and 4.

The boundary-layer parameters (displacement and momentum thicknesses) are also used to calculate the conditions downstream of the rotor after the flow has mixed to a uniform state. The procedure described in reference 5 is used for this purpose.

This report presents a description of the input and output and a complete FORTRAN IV listing of the program. A brief description of the computer program and method of design is also given. An example of the program output is included to indicate the use of the program and the results obtainable.

METHOD OF ANALYSIS

The design of two-dimensional supersonic rotor blades that are corrected for boundary-layer displacement thickness is described herein. The ideal rotor passage is designed by the method of characteristics as applied to the isentropic flow of a perfect gas. Boundary-layer parameters (displacement and momentum thicknesses) are then calculated for the ideal blades. The final rotor blade profile is obtained by adding the displacement thicknesses to the ideal rotor coordinates.

Rotor Blade Description and Design

The design of the ideal blade passage is based on establishing vortex flow within the passage by a procedure analogous to that given in reference 2. A typical passage is shown in figure 1. The passage consists essentially of three major parts: (1) inlet transition arcs, (2) circular arcs, and (3) outlet transition arcs. The inlet transition arcs (upper and lower) are required to convert the assumed uniform parallel flow at the passage inlet into vortex flow. The concentric circular arcs turn and maintain the vortex flow condition. The outlet transition arcs reconvert the vortex flow into uniform parallel flow at the passage exit. Straight-line segments parallel to the inlet and outlet flow direction complete the ideal passage.

As seen from figure 1, the ideal passage is designed so that the outlet spacing is less than the inlet spacing. This is necessary if the corrected passage (i. e., the passage corrected by the boundary-layer displacement thicknesses) is to have equal outlet and inlet spacing (see fig. 2). For an ideal passage designed for impulse conditions (equal inlet and outlet Mach numbers), this is accomplished by having less circular turning for the outlet portion of the passage. That is, the outlet circular arcs JK and DE (fig. 1) are less than the corresponding inlet circular arcs IJ and CD. An iterative

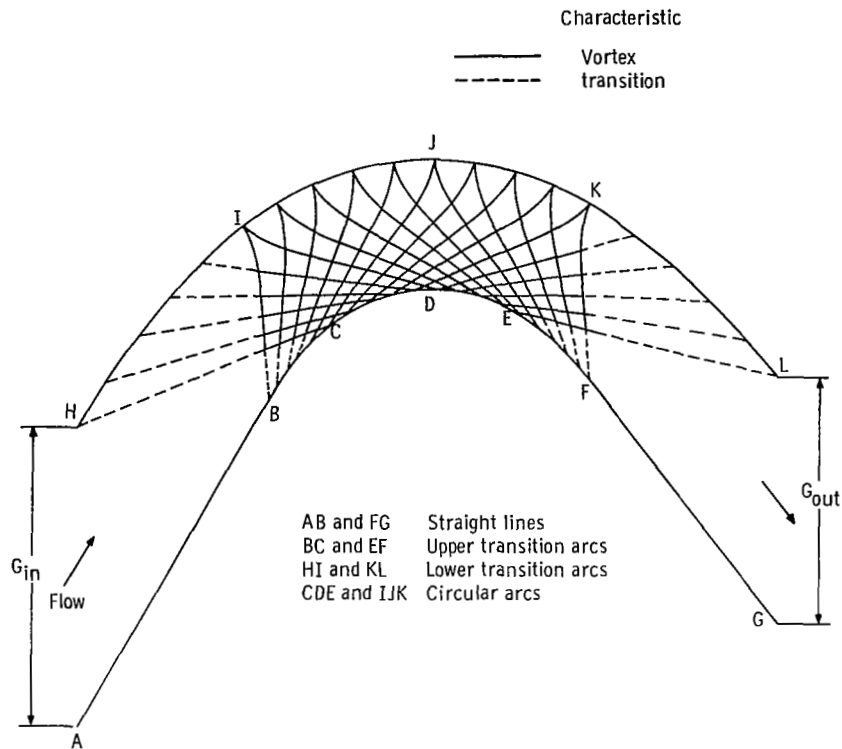


Figure 1. - Design of ideal supersonic flow passage by method of characteristics.

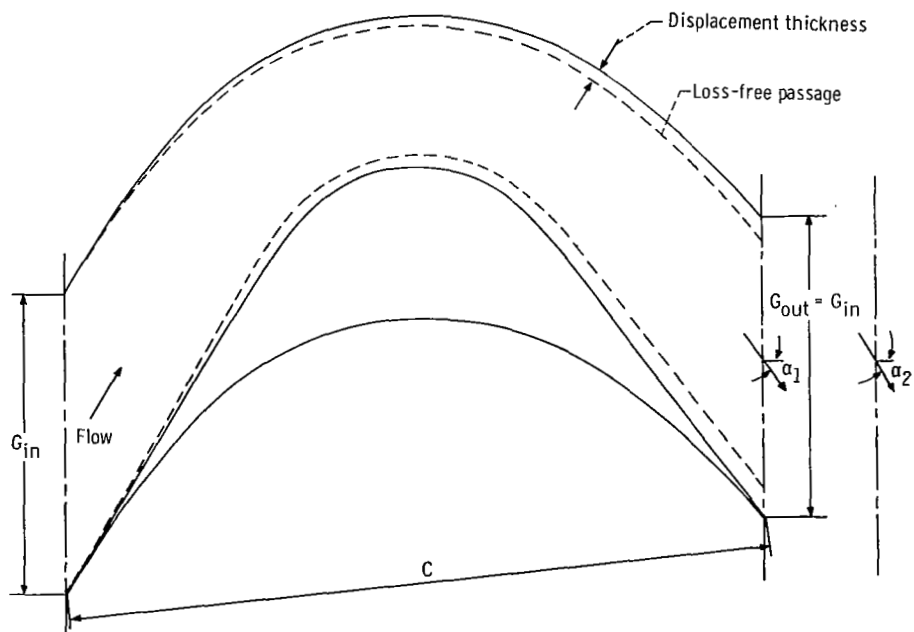


Figure 2. - Design of supersonic rotor blade section.

procedure performed by the computer program determines the condition for which the corrected passage has equal inlet and outlet spacings.

Boundary-Layer Calculations

The boundary-layer parameters (displacement and momentum thicknesses) are calculated for the ideal rotor passage by using the computer program described in 4. The program uses Cohen and Reshotko's method (ref. 6) for the calculation of laminar boundary layers and Sasman and Cresci's method (ref. 7) for turbulent boundary layers. Curvature effects are not considered in these calculations.

In the laminar regime, a single ordinary differential equation (the momentum integral equation) is solved numerically. The results of this method (as explained in ref. 6) have to be extended for flows in highly favorable pressure gradients as might occur over some portions of the rotor blade. For turbulent flow, coupled first-order ordinary differential equations (the momentum and moment-of-momentum integral equations) are solved using Runge-Kutta techniques.

The displacement thicknesses obtained from the program are then added to the ideal rotor blade coordinates to obtain the corrected blade profile. As discussed previously, an iterative procedure is performed by the computer program to determine the conditions for which the corrected passage has equal inlet and outlet spacings. It should be noted that it is not always possible to obtain a blade design by this procedure because of boundary-layer separation or large boundary-layer growth.

A number of options related to the boundary-layer calculations are also available to the user. Because of the adverse pressure gradients that exist within the rotor passage, the computer program will generally predict laminar separation at these locations. The program allows for the reattachment of the flow and continuation of the calculations for turbulent flow, if this is desired. The user can also, if he wishes, force transition to turbulent flow at any point of the calculations, including the inlet.

Aftermixing Conditions

The displacement and momentum thickness at the rotor exit (station 1, fig. 2) can be used to calculate the aftermixing conditions downstream of the rotor assuming that the flow mixes to a uniform state. Application of the continuity, momentum, and energy equations between stations 1 and 2 (fig. 2) results in the determination of aftermixing Mach number, flow angle, pressure ratio, and kinetic energy loss. The calculation procedure has been described in reference 5.

Subsonic and supersonic aftermixing axial Mach number solutions are possible for this loss model when the free-stream axial Mach number at the blade exit (before mixing) is supersonic. The subsonic solution corresponds to mixing plus oblique shock losses, whereas the supersonic solution corresponds to shockless mixing. A more detailed discussion of the different solutions can be found in reference 8.

DESCRIPTION OF INPUT

A description of the input for the FORTRAN IV computer program is given in this section. The input consists primarily of rotor inlet and outlet Mach number, upper- and lower-surface Mach number, inlet flow angle, specific-heat ratio, and total flow conditions. Either U.S. customary units or the International System of units may be used for input. This option is controlled by the input variable KEM. The program gas properties are set up for air. For gases other than air the changes required to the program are described in appendix A.

The input format is shown in table I. The input variables are

BETAN	inlet flow angle, β_i , deg
VIN	inlet Prandtl-Meyer angle, ν_i , deg
VLOW	lower-surface Prandtl-Meyer angle, ν_l , deg
VUP	upper-surface Prandtl-Meyer angle, ν_u , deg
VOU	outlet Prandtl-Meyer angle, ν_o , deg
BETAT	initial estimate of outlet flow angle, β_o , deg (in the absence of a better estimate use BETAN)
DELV	flow-turning increment (recommended value, 0.1), $\Delta\nu$, deg (The flow-turning increment must be specified so that $(\nu_i - \nu_l)/\Delta\nu$, $(\nu_o - \nu_l)/\Delta\nu$, $(\nu_u - \nu_i)/\Delta\nu$, and $(\nu_u - \nu_o)/\Delta\nu$ are all integers.)
GAM	specific-heat ratio, γ
NTURBU	integer number of station on upper surface, if any, at which user wishes turbulent boundary layer to begin (If NTURBU is set equal to zero, the program will begin laminar boundary-layer calculations at the rotor inlet. Value of CTHETU must be specified; see section Instructions for Preparing Input. NTURBU cannot be set equal to 1 because initial values of displacement and momentum thicknesses are not available to program.)
NTURBL	same as NTURBU except for lower surface

TABLE I. - INPUT FORM

[Numbers in corners are card column numbers.]

1	6		9	14		17	22	25	30	33	38	41	46	49	54	57	62		
BETAN		VIN		VLOW		VUP		VOUT		BETAT		DELV		GAM					
1	3	4	6	7	9	11	19	21	29	31	39	41	49	51	59	61	69	71	79
NURBL		NVP		R		PTZ		TTZ		XMAX		TE		CTHETU		CTHETL			
1	5	6	10	11	15	16	20	21	25	26	30	31	35						
KPRE		KGRAD		KSDE		KLAM		KMAIN		KPROF		KEM							

NVP	integer number of points desired in velocity profile of the boundary layer at each station (must have at least 1)
R	gas constant, J/(kg)(K); (ft)(lbf)/(slug)(^o R)
PTZ	inlet relative total pressure, N/m ² ; lbf/ft ²
TTZ	inlet relative total temperature, K; ^o R
XMAX	blade chord, m; ft
TE	thickness of trailing edge (used only for aftermixing calculations), m; ft
CTHETU	real variable indicating ratio of momentum thickness after reattachment to momentum thickness at laminar separation for upper surface
CTHETL	same as CTHETU except for lower surface
KPRE	integer (0 or 1) indicating whether printing of output from PRECAL is desired: Output suppressed 0 Output printed 1
KGRAD	integer (0 or 1, see KPRE) indicating whether printing of surface gradients of velocity and Mach number is desired
KSDE	integer (0 or 1, see KPRE) indicating whether printing of solutions of laminar and turbulent differential equations is desired
KLAM	integer (0 or 1, see KPRE) indicating whether printing of laminar calculations for location of instability and transition is desired
KMAIN	integer (0 or 1, see KPRE) indicating whether printing of principal calculated boundary-layer parameters is desired
KPROF	integer (0 or 1, see KPRE) indicating whether printing of velocity profiles is desired
KEM	integer (0 or 1) indicating which of two allowable sets of units are used in input: U. S. customary (pounds force, slugs, feet, seconds, degrees Rankine, and foot-pounds) 0 International System (newtons, kilograms, meters, seconds, kelvin, and joules) 1

Instructions for Preparing Input

Laminar separation and reattachment. - If NTURBU and NTURBL are set equal to zero, the program will begin laminar-boundary-layer calculations at the rotor inlet. Because of the adverse pressure gradients that exist on the upper and lower surfaces the program may predict separation on these surfaces. The program is set up so that reattachment of the flow is allowed to occur, and the calculations proceed for a turbulent boundary layer, if this is desired. Values for CTHETU and CTHETL (different from zero), which are the ratios of the momentum thickness after reattachment to the momentum thickness at laminar separation, must be specified as input. The values of CTHETU and CTHETL can be set equal to 1, which is essentially equivalent to assuming that transition to turbulent flow occurs at the point of imminent laminar separation. If CTHETU and CTHETL are set equal to zero, reattachment will not occur.

Output. - Usually KPRE, KGRAD, KSDE, KLAM, and KPROF are set equal to zero. They do not give the main output of the boundary-layer section. However, if this additional output is desired, these quantities are set equal to 1. A description of this output is given in appendix B.

DESCRIPTION OF MAIN OUTPUT

An example of the output from the program is shown in table II. The output is given in U. S. customary units, and each section has been numbered to correspond to the following description:

(1) Output 1 of the program is a listing of the input data used for the rotor design plus a listing of miscellaneous parameters including

- (a) Inlet, outlet, and surface critical velocity ratios M^* and Mach numbers M .
- (b) Dimensionless blade spacing G^* , chord C^* , and solidity $SIGMA$.

For each surface (lower and upper) items (2) to (5) are given for each iteration (if more than one iteration is required).

(2) Output 2 gives the input needed by the boundary-layer program. The first two lines of this output are a listing of the input data except for NST (which is the total number of station points) and UPMACH (which is the inlet relative Mach number). The remaining output is the surface coordinates and velocity distribution:

XOM dimensionless X-coordinate, X/C^*

YOM dimensionless Y-coordinate, Y/C^*

VVCR critical velocity ratio

TWAL wall temperature, K; $^{\circ}R$ (wall temperature assumed equal to total temperature TTZ)

TABLE II. - EXAMPLE OF PROGRAM OUTPUT

DESIGN PARAMETERS

BETA(IN) = 70.0000 DEG V(IN) = 39.1200 DEG V(UP) = 49.1200 DEG V(DUT) = 39.1200 DEG BETA(OUT) = -61.1300 DEG
 DELTA γ = 0.1000 DEG V(LCW) = 19.1200 DEG GAMMA = 1.4000

MISCELLANEOUS PARAMETERS

M*(IN) = 1.0257 M(IN) = 2.4999 M(DUT) = 2.4999 M*(OUT) = 1.0257
 M*(LOW) = 1.5068 M(LCW) = 1.7448 M(UP) = 2.9572 M*(UP) = 1.9562

INPJT BETA(0) = -61.1300 DEG

G*(IN) = 0.5944 C* = 1.6406 SIGMA(IN) = 2.7603
 G*(DUT) = 0.4210 SIGMA(OUT) = 3.0966

*** UPPER SURFACE ***

BOUNDARY LAYER - INPUT

YST VVP GAM H PTZ TIZ UPMAC1 XMAX VTURB CTHETA TE
 64 5 1.400 1714.70 676.00 1913.30 2.4999 0.04183 3 1.000 0.00040

KPRE KGRAD KSDE KLAM KMAIN KPRDF KEM
 0 0 0 0 1 0 0

XDM	YDM	VVCR	TWAL
-0.49257	-0.43870	1.02569	1913.30
-0.47711	-0.38798	1.02569	1913.30
-0.45365	-0.33725	1.02569	1913.30
-0.44013	-0.28652	1.02569	1913.30
-0.42172	-0.23579	1.02569	1913.30
-0.40325	-0.18505	1.02569	1913.30
-0.38479	-0.13433	1.02569	1913.30
-0.36633	-0.08361	1.02569	1913.30
-0.34787	-0.03288	1.02569	1913.30
-0.32940	0.01785	1.02569	1913.30
-0.31094	0.06858	1.02569	1913.30
-0.30783	0.07688	1.03951	1913.30
-0.30453	0.08524	1.05316	1913.30
-0.30122	0.09367	1.06663	1913.30
-0.29791	0.10218	1.07993	1913.30
-0.29459	0.11080	1.09305	1913.30
-0.28921	0.11952	1.10601	1913.30
-0.28779	0.12836	1.11891	1913.30
-0.28010	0.13734	1.13143	1913.30
-0.27513	0.14646	1.14390	1913.30
-0.26985	0.15580	1.15620	1913.30
-0.26524	0.16524	1.16820	1913.30
-0.25870	0.17472	1.18000	1913.30
-0.25370	0.18420	1.19160	1913.30
-0.24833	0.19368	1.20300	1913.30
-0.24250	0.20316	1.21430	1913.30
-0.23620	0.21264	1.22550	1913.30
-0.22940	0.22212	1.23660	1913.30
-0.22210	0.23160	1.24760	1913.30
-0.21430	0.24108	1.25850	1913.30
-0.20600	0.25056	1.26930	1913.30
-0.19720	0.26004	1.28000	1913.30
-0.18790	0.26952	1.29060	1913.30
-0.17810	0.27900	1.30110	1913.30
-0.16780	0.28848	1.31150	1913.30
-0.15700	0.29796	1.32180	1913.30
-0.14570	0.30744	1.33200	1913.30
-0.13390	0.31692	1.34210	1913.30
-0.12160	0.32640	1.35210	1913.30
-0.10880	0.33588	1.36200	1913.30
-0.09550	0.34536	1.37180	1913.30
-0.08170	0.35484	1.38150	1913.30
-0.06740	0.36432	1.39110	1913.30
-0.05260	0.37380	1.40060	1913.30
-0.03730	0.38328	1.41000	1913.30
0.02150	0.39276	1.41930	1913.30
0.04680	0.40224	1.42850	1913.30
0.07210	0.41172	1.43760	1913.30
0.09740	0.42120	1.44660	1913.30
0.12270	0.43068	1.45550	1913.30
0.14800	0.44016	1.46430	1913.30
0.17330	0.44964	1.47300	1913.30
0.19860	0.45912	1.48160	1913.30
0.22390	0.46860	1.49010	1913.30
0.24920	0.47808	1.49850	1913.30
0.27450	0.48756	1.50680	1913.30
0.30000	0.49704	1.51500	1913.30
0.32530	0.50652	1.52310	1913.30
0.35060	0.51600	1.53110	1913.30
0.37590	0.52548	1.53900	1913.30
0.40120	0.53496	1.54680	1913.30
0.42650	0.54444	1.55450	1913.30
0.45180	0.55392	1.56210	1913.30
0.47710	0.56340	1.56960	1913.30
0.50240	0.57288	1.57700	1913.30
0.52770	0.58236	1.58430	1913.30
0.55300	0.59184	1.59150	1913.30
0.57830	0.60132	1.59860	1913.30
0.60360	0.61080	1.60560	1913.30
0.62890	0.62028	1.61250	1913.30
0.65420	0.62976	1.61930	1913.30
0.67950	0.63924	1.62600	1913.30
0.70480	0.64872	1.63260	1913.30
0.73010	0.65820	1.63910	1913.30
0.75540	0.66768	1.64550	1913.30
0.78070	0.67716	1.65180	1913.30
0.80600	0.68664	1.65800	1913.30
0.83130	0.69612	1.66410	1913.30
0.85660	0.70560	1.67010	1913.30
0.88190	0.71508	1.67600	1913.30
0.90720	0.72456	1.68180	1913.30
0.93250	0.73404	1.68750	1913.30
0.95780	0.74352	1.69310	1913.30
0.98310	0.75300	1.69860	1913.30
1.00840	0.76248	1.70400	1913.30
1.03370	0.77196	1.70930	1913.30
1.05900	0.78144	1.71450	1913.30
1.08430	0.79092	1.71960	1913.30
1.10960	0.80040	1.72460	1913.30
1.13490	0.80988	1.72950	1913.30
1.16020	0.81936	1.73430	1913.30
1.18550	0.82884	1.73900	1913.30
1.21080	0.83832	1.74360	1913.30
1.23610	0.84780	1.74810	1913.30
1.26140	0.85728	1.75250	1913.30
1.28670	0.86676	1.75680	1913.30
1.31200	0.87624	1.76100	1913.30
1.33730	0.88572	1.76510	1913.30
1.36260	0.89520	1.76910	1913.30
1.38790	0.90468	1.77300	1913.30
1.41320	0.91416	1.77680	1913.30
1.43850	0.92364	1.78050	1913.30
1.46380	0.93312	1.78410	1913.30
1.48910	0.94260	1.78760	1913.30
1.51440	0.95208	1.79100	1913.30
1.53970	0.96156	1.79430	1913.30
1.56500	0.97104	1.79750	1913.30
1.59030	0.98052	1.80060	1913.30
1.61560	0.99000	1.80360	1913.30
1.64090	0.99948	1.80650	1913.30
1.66620	1.00896	1.80930	1913.30
1.69150	1.01844	1.81200	1913.30
1.71680	1.02792	1.81460	1913.30
1.74210	1.03740	1.81710	1913.30
1.76740	1.04688	1.81950	1913.30
1.79270	1.05636	1.82180	1913.30
1.81800	1.06584	1.82400	1913.30
1.84330	1.07532	1.82610	1913.30
1.86860	1.08480	1.82810	1913.30
1.89390	1.09428	1.83000	1913.30
1.91920	1.10376	1.83180	1913.30
1.94450	1.11324	1.83350	1913.30
1.96980	1.12272	1.83510	1913.30
1.99510	1.13220	1.83660	1913.30
2.02040	1.14168	1.83800	1913.30
2.04570	1.15116	1.83930	1913.30
2.07100	1.16064	1.84050	1913.30
2.09630	1.17012	1.84160	1913.30
2.12160	1.17960	1.84260	1913.30
2.14690	1.18908	1.84350	1913.30
2.17220	1.19856	1.84430	1913.30
2.19750	1.20804	1.84500	1913.30
2.22280	1.21752	1.84560	1913.30
2.24810	1.22700	1.84610	1913.30
2.27340	1.23648	1.84650	1913.30
2.29870	1.24596	1.84680	1913.30
2.32400	1.25544	1.84700	1913.30
2.34930	1.26492	1.84710	1913.30
2.37460	1.27440	1.84710	1913.30
2.40000	1.28388	1.84700	1913.30
2.42530	1.29336	1.84680	1913.30
2.45060	1.30284	1.84650	1913.30
2.47590	1.31232	1.84600	1913.30
2.50120	1.32180	1.84540	1913.30
2.52650	1.33128	1.84470	1913.30
2.55180	1.34076	1.84390	1913.30
2.57710	1.35024	1.84300	1913.30
2.60240	1.35972	1.84200	1913.30
2.62770	1.36920	1.84090	1913.30
2.65300	1.37868	1.83970	1913.30
2.67830	1.38816	1.83840	1913.30
2.70360	1.39764	1.83700	1913.30
2.72890	1.40712	1.83550	1913.30
2.75420	1.41660	1.83390	1913.30
2.77950	1.42608	1.83220	1913.30
2.80480	1.43556	1.83040	1913.30
2.83010	1.44504	1.82850	1913.30
2.85540	1.45452	1.82650	1913.30
2.88070	1.46400	1.82440	1913.30
2.90600	1.47348	1.82220	1913.30
2.93130	1.48296	1.81990	1913.30
2.95660	1.49244	1.81750	1913.30
2.98190	1.50192	1.81500	1913.30
3.00720	1.51140	1.81240	1913.30
3.03250	1.52088	1.80970	1913.30
3.05780	1.53036	1.80690	1913.30
3.08310	1.53984	1.80400	1913.30
3.10840	1.54932	1.80100	1913.30
3.13370	1.55880	1.79790	1913.30
3.15900	1.56828	1.79470	1913.30
3.18430	1.57776	1.79140	1913.30
3.20960	1.58724	1.78800	1913.30
3.23490	1.59672	1.78450	1913.30
3.26020	1.60620	1.78090	1913.30
3.28550	1.61568	1.77720	1913.30
3.31080	1.62516	1.77340	1913.30
3.33610	1.63464	1.76950	1913.30
3.36140	1.64412	1.76550	1913.30
3.38670	1.65360	1.76140	1913.30
3.41200	1.66308	1.75720	1913.30
3.43730	1.67256	1.75290	1913.30
3.46260	1.68204	1.74840	1913.30
3.48790	1.69152	1.74380	1913.30
3.51320	1.70100	1.73910	1913.30
3.53850	1.71048	1.73430	1913.30
3.56380	1.71996	1.72940	1913.30
3.58910	1.72944	1.72440	1913.30
3.61440	1.73892	1.71930	1913.30
3.63970	1.74840	1.71410	1913.30
3.66500	1.75788	1.70880	1913.30
3.69030	1.76736	1.70340	1913.30
3.71560	1.77684	1.69790	1913.30
3.74090	1.78632	1.69230	1913.30
3.76620	1.79580	1.68660	1913.30
3.79150	1.80528	1.68080	1913.30
3.81680	1.81476	1.67490	1913.30
3.84210	1.82424	1.66890	1913.30
3.86740	1.83372	1.66280	1913.30
3.89270	1.84320	1.65660	1913.30
3.91800	1.85268	1.65030	1913.30
3.94330	1.86216	1.64390	1913.30
3.96860	1.87164	1.63740	1913.30
3.99390	1.88112	1.63080	1913.30
4.01920	1.89060	1.62410	1913.30
4.04450	1.90008	1.61730	1913.30
4.06980	1.90956	1.61040	1913.30
4.09510	1.91904	1.60340	1913.30
4.12040	1.92852	1.59630	1913.30
4.14570	1.93800	1.58910	1913.30
4.17100	1.94748	1.58180	1913.30
4.19630	1.95696	1.57440	1913.30
4.22160	1.96644	1.56690	1913.30
4.24690	1.97592	1.55930	1913.30
4.27220	1.98540	1.55160	1913.30
4.29750	1.99488	1.54380	1913.30

TABLE II. - Continued. EXAMPLE OF PROGRAM OUTPUT

0.28774	0.13117	1.85316	1913.30
0.29223	0.12342	1.83951	1913.30
0.29655	0.11570	1.82569	1913.30
0.31712	0.07656	1.82569	1913.30
0.33763	0.04142	1.82569	1913.30
0.35409	0.00428	1.82569	1913.30
0.37855	-0.03287	1.82569	1913.30
0.39974	-0.07001	1.82569	1913.30
0.41952	-0.10715	1.82569	1913.30
0.44003	-0.14430	1.82569	1913.30
0.46047	-0.18144	1.82569	1913.30
0.48095	-0.21858	1.82569	1913.30
0.50143	-0.25572	1.82569	1913.30

PRELIMINARY CALCULATIONS

PSZ	=	39.62063
TSZ	=	800.4114
UZ	=	3571.79858
ASZ	=	1428.8039
ATZ	=	2143.1355
RHSZ	=	0.2717091E-04
RHTZ	=	0.2062952E-03
MUSZ	=	0.5372009E-06
MUTZ	=	0.9149355E-06
NUSZ	=	0.1977118E-01
NUTZ	=	0.4435079E-02
CP	=	6001.4498
PR	=	0.67232
TC	=	0.4892751E-02
ARCL	=	0.0737

STATION	PRES	UE	MF	PUPTZ	VOVCR
1	39.62063	3571.79855	2.499452	0.058541	1.325694
2	39.62063	3571.79855	2.499852	0.058541	1.325694
3	39.62063	3571.79855	2.499852	0.058541	1.325694
4	39.62063	3571.79855	2.499852	0.058541	1.325694
5	39.62063	3571.79855	2.499852	0.058541	1.325694
6	39.62063	3571.79855	2.499852	0.058541	1.325694
7	39.62063	3571.79855	2.499852	0.058541	1.325694
8	39.62063	3571.79855	2.499852	0.058541	1.325694
9	39.62063	3571.79855	2.499852	0.058541	1.325694
10	39.62063	3571.79855	2.499852	0.058541	1.325694
11	39.62063	3571.79855	2.499852	0.058541	1.325694
12	37.04862	3598.83670	2.543043	0.094741	1.339514

PRINCIPAL BOUNDARY LAYER INFORMATION

INSTABILITY DOES NOT OCCUR

TRANSITION OCCURS AT STATION 3

SEPARATION DOES NOT OCCUR

LAMINAR BOUNDARY LAYER - STATIONS 1 TO 2

TURBULENT BOUNDARY LAYER - STATIONS 3 TO 12

DIMENSIONED BLADE COORDINATES

ANGLES OF ROTATION (DEG)

STATION	X	Y	YCDJR	YTRAN	
1	-0.02073	-0.01835	-0.01835	-0.03351	70.0000
2	-0.01994	-0.01623	-0.01753	-0.03278	70.0000
3	-0.01919	-0.01411	-0.01523	-0.03039	70.0000
4	-0.01841	-0.01199	-0.01325	-0.02841	70.0000
5	-0.01764	-0.00986	-0.01125	-0.02640	70.0000
6	-0.01687	-0.00774	-0.00924	-0.02439	70.0000
7	-0.01610	-0.00562	-0.00721	-0.02237	70.0000
8	-0.01532	-0.00350	-0.00518	-0.02034	70.0000
9	-0.01455	-0.00138	-0.00315	-0.01830	70.0000
10	-0.01378	0.00077	-0.00111	-0.01627	70.0000
11	-0.01301	0.00287	0.00098	-0.01417	70.0000
12	-0.01228	0.00327	0.00135	-0.01381	69.0000

TABLE II. - Continued. EXAMPLE OF PROGRAM OUTPUT

STATION	X	S	DELSR	THET	DELTA	FJRM	FJRM1
1	-0.020730	0.	-0.	-0.	-0.	7.4563	2.8596
2	-0.019958	0.002258	0.000479	0.000064	0.000826	7.4563	2.8566
3	-0.019185	0.004516	0.000386	0.000091	0.002314	4.2447	1.4000
4	-0.018413	0.006774	0.000433	0.000099	0.002331	4.3709	1.4561
5	-0.017641	0.009033	0.000474	0.000106	0.002397	4.4554	1.4936
6	-0.016868	0.011291	0.000512	0.000113	0.002490	4.5081	1.5171
7	-0.016096	0.013549	0.000545	0.000120	0.002600	4.5393	1.5309
8	-0.015324	0.015807	0.000577	0.000127	0.002719	4.5567	1.5367
9	-0.014551	0.018065	0.000607	0.000133	0.002844	4.5656	1.5426
10	-0.013779	0.020323	0.000636	0.000139	0.002973	4.5691	1.5442
11	-0.013007	0.022582	0.000666	0.000143	0.003100	4.5296	1.5266
12	-0.012236	0.022952	0.000669	0.000146	0.003324	4.5825	1.5041

STATION	CF	TAUW	RTH	DTDY	NUSS	HTRAN	CRN
1	0.	0.	-0.	0.	0.	0.	2.021
2	0.06588	5.07481	3.0	-261125.21	3.08	-1277.6206	2.021
3	0.00768	1.33144	4.3	-78495.36	2.69	-384.0582	0.608
4	0.00688	1.19193	4.7	-70270.32	3.51	-343.8152	0.508
5	0.00636	1.10244	5.0	-64994.51	4.45	-318.0020	0.608
6	0.00603	1.04491	5.3	-61632.55	5.28	-301.4064	0.608
7	0.00581	1.00701	5.7	-59368.07	6.11	-290.4732	0.508
8	0.00566	0.98103	6.0	-57836.59	6.94	-282.9800	0.508
9	0.00555	0.96238	6.3	-56737.07	7.78	-277.6003	0.608
10	0.00547	0.94835	6.6	-55909.82	8.52	-273.5528	0.608
11	0.00559	0.96822	6.7	-57081.50	9.78	-279.2856	0.608
12	0.00572	0.95860	6.5	-56942.25	9.77	-278.6043	0.508

*** LOWER SURFACE ***

BOUNDARY LAYER - INPUT

NST	VVP	GAM	R	PTZ	TTZ	UPMACH	XMAX	NTURB	CTHETA	TE
60	5	1.400	1714.70	676.80	1913.30	2.4999	0.04183	3	1.000	0.00040

KPRE	KGRAD	KSDR	KLAM	KMAIN	KPROF	KEM
0	0	0	0	1	0	0

XOM	YOM	VVCR	THAL
-0.49557	-0.07642	1.82569	1913.30
-0.48597	-0.05078	1.81169	1913.30
-0.47637	-0.02653	1.79752	1913.30
-0.46685	-0.00355	1.78316	1913.30
-0.45735	0.01824	1.76861	1913.30
-0.44793	0.03895	1.75388	1913.30
-0.43943	0.05865	1.73695	1913.30
-0.42991	0.07740	1.72383	1913.30
-0.41979	0.09530	1.70850	1913.30
-0.41049	0.11238	1.69297	1913.30
-0.40123	0.12871	1.67724	1913.30
-0.39200	0.14434	1.66129	1913.30
-0.38281	0.15932	1.64511	1913.30
-0.37353	0.17369	1.62871	1913.30
-0.36442	0.18749	1.61208	1913.30
-0.35535	0.20075	1.59520	1913.30
-0.34624	0.21350	1.57807	1913.30
-0.33714	0.22578	1.56069	1913.30
-0.32805	0.23761	1.54303	1913.30
-0.31895	0.24902	1.52509	1913.30
-0.30983	0.26002	1.50685	1913.30
-0.28604	0.28604	1.50685	1913.30
-0.26032	0.30988	1.50685	1913.30
-0.23202	0.33136	1.50685	1913.30
-0.20225	0.35032	1.50685	1913.30
-0.17095	0.36662	1.50685	1913.30
-0.13835	0.38012	1.50685	1913.30
-0.10472	0.39073	1.50685	1913.30
-0.07024	0.39837	1.50685	1913.30
-0.03525	0.40298	1.50685	1913.30
0.00000	0.40451	1.50685	1913.30
0.03525	0.40298	1.50685	1913.30
0.07024	0.39837	1.50685	1913.30
0.10472	0.39073	1.50685	1913.30
0.13835	0.38012	1.50685	1913.30
0.17095	0.36662	1.50685	1913.30
0.20225	0.35032	1.50685	1913.30
0.23202	0.33136	1.50685	1913.30
0.26032	0.30988	1.50685	1913.30
0.28604	0.28604	1.50685	1913.30
0.27675	0.29522	1.52509	1913.30
0.28743	0.28530	1.54303	1913.30
0.29833	0.27507	1.56069	1913.30
0.30913	0.26434	1.57807	1913.30
0.32015	0.25314	1.59520	1913.30
0.33122	0.24143	1.61208	1913.30
0.34233	0.22923	1.62871	1913.30
0.35355	0.21644	1.64511	1913.30
0.36505	0.20306	1.66129	1913.30
0.37653	0.18994	1.67724	1913.30

TABLE II. - Continued. EXAMPLE OF PROGRAM OUTPUT

0.38825	0.17433	1.69297	1913.30
0.40035	0.15888	1.70850	1913.30
0.41234	0.14264	1.72383	1913.30
0.42427	0.12555	1.73995	1913.30
0.43553	0.10755	1.75388	1913.30
0.44737	0.08855	1.76861	1913.30
0.46182	0.06848	1.78316	1913.30
0.47473	0.04724	1.79752	1913.30
0.48793	0.02476	1.81169	1913.30
0.50143	0.00091	1.82569	1913.30

PRELIMINARY CALCULATIONS

PSZ	=	39.62063
TSZ	=	850.4114
UZ	=	3571.79858
ASZ	=	1428.8039
ATZ	=	2143.1355
RHSZ	=	0.2717091E-04
RHTZ	=	0.2062952E-03
MUSZ	=	0.3372009E-06
MUTZ	=	0.9149355E-06
NUSZ	=	0.1977118E-01
NUTZ	=	0.4435079E-02
CP	=	6001.4498
PR	=	0.67232
TC	=	0.4892751E-02
ARCL	=	0.0592

STATION	PRES	UE	ME	POPTZ	WQVCR
1	39.62063	3571.79855	2.499852	0.058541	1.325694
2	42.33229	3544.40982	2.457333	0.062548	1.311694
3	45.18726	3516.68137	2.415483	0.066766	1.797521
4	48.19350	3488.58292	2.374237	0.071208	1.783159
5	51.35440	3460.12845	2.333597	0.075878	1.768615
6	54.57762	3431.29196	2.293511	0.080789	1.753875
7	58.15675	3402.09058	2.253987	0.085944	1.738949
8	61.82872	3372.50253	2.214980	0.091354	1.723825
9	65.66948	3342.52209	2.176470	0.097029	1.708501
10	69.69449	3312.14563	2.138441	0.102975	1.592975
11	73.91054	3281.35953	2.100862	0.109206	1.577239
12	78.32432	3250.15289	2.063711	0.115727	1.561287

PRINCIPAL BOUNDARY LAYER INFORMATION

INSTABILITY DOES NOT OCCUR

TRANSITION OCCURS AT STATION 3

SEPARATION DOES NOT OCCUR

LAMINAR BOUNDARY LAYER - STATIONS 1 TO 2

TURBULENT BOUNDARY LAYER - STATIONS 3 TO 60

DIMENSIONED BLADE COORDINATES

ANGLES OF ROTATION (DEG)

STATION	X	Y	YCORR	YTRAV	
1	-0.02073	-0.00320	-0.00320	-0.01835	70.0000
2	-0.02033	-0.00212	-0.00121	-0.01636	69.0000
3	-0.01993	-0.00111	-0.00043	-0.01559	68.0000
4	-0.01953	-0.00015	0.00055	-0.01459	67.0000
5	-0.01913	0.00076	0.00150	-0.01356	66.0000
6	-0.01874	0.00163	0.00233	-0.01278	65.0000
7	-0.01834	0.00245	0.00321	-0.01194	64.0000
8	-0.01795	0.00324	0.00400	-0.01116	63.0000
9	-0.01756	0.00399	0.00475	-0.01040	62.0000
10	-0.01717	0.00470	0.00547	-0.00959	61.0000
11	-0.01678	0.00536	0.00613	-0.00900	60.0000
12	-0.01640	0.00604	0.00681	-0.00835	59.0000

TABLE II. - Concluded. EXAMPLE OF PROGRAM OUTPUT

STATION	X	S	DELTA	THET	DELTA	FORM	FORM1
1	-0.020730	0.	-0.	-0.	-0.	7.4563	2.8586
2	-0.020328	0.00145	0.000324	0.000045	0.000573	7.2629	2.8413
3	-0.019927	0.002236	0.000254	0.000063	0.001525	4.0559	1.4000
4	-0.019528	0.003277	0.000278	0.000067	0.001435	4.1361	1.4800
5	-0.019131	0.004271	0.000298	0.000071	0.001381	4.1941	1.5509
6	-0.018735	0.005223	0.000314	0.000075	0.001348	4.2296	1.6121
7	-0.018342	0.006137	0.000332	0.000078	0.001329	4.2462	1.6646
8	-0.017950	0.007014	0.000346	0.000081	0.001319	4.2482	1.7104
9	-0.017559	0.007958	0.000359	0.000085	0.001313	4.2403	1.7512
10	-0.017171	0.0088671	0.000371	0.000088	0.001311	4.2263	1.7870
11	-0.016783	0.009456	0.000383	0.000091	0.001312	4.2094	1.8251
12	-0.016397	0.010216	0.000395	0.000094	0.001314	4.1921	1.8609

STATION	CF	TAUM	RT	DTDY	NUSS	HTRAV	CRV
1	0.	0.	-0.	0.	0.	0.	2.021
2	0.08368	6.78206	2.3	-374773.63	2.28	-1833.6740	1.847
3	0.00859	1.54670	3.3	-91984.48	1.51	-450.0371	0.631
4	0.00747	1.43555	3.8	-81970.20	2.14	-401.0598	0.643
5	0.00664	1.29963	4.2	-74224.09	2.55	-363.1600	0.655
6	0.00597	1.20582	4.7	-68292.74	2.93	-334.1394	0.667
7	0.00548	1.13438	5.1	-63599.37	3.27	-311.6676	0.678
8	0.00508	1.07854	5.7	-60037.27	3.58	-293.7474	0.690
9	0.00474	1.03290	6.2	-56985.88	3.88	-278.8177	0.702
10	0.00445	0.99348	6.8	-54313.07	4.15	-265.7403	0.714
11	0.00419	0.95733	7.4	-51849.98	4.41	-253.6890	0.726
12	0.00395	0.9221	8.0	-49473.16	4.53	-242.0599	0.739

AFTERMIXING PROPERTIES

ROTOR WITH NO BOUNDARY LAYER CORRECTION

XMF51 = 2.4999 SPACING = 0.010735 TE = 0.00040 XW2 = 1.5823 V/VCR1 = 1.825 XMX1 = 1.207 XMX2 = 0.641

ALPH1 = 61.130 ALPH2 = 66.110 PT2/PT0 = 0.2821 PT0/P2 = 14.678 T2/T0 = 0.6653 V/VCR2 = 1.415 EBAR = 0.3773 ETA-V = 0.6227

ROTOR WITH BOUNDARY LAYER CORRECTION

XMF51 = 2.4999 SPACING = 0.015225 TE = 0.00040 XW2 = 1.8334 V/VCR1 = 1.525 XMX1 = 1.207 XMX2 = 0.710

ALPH1 = 61.130 ALPH2 = 67.221 PT2/PT0 = 0.4543 PT0/P2 = 13.310 T2/T0 = 0.5980 V/VCR2 = 1.553 EBAR = 0.23088 ETA-V = 0.7091

SUPERSONIC SOLUTION

XMF51 = 2.4999 SPACING = 0.015225 TE = 0.00040 XW2 = 2.4299 V/VCR1 = 1.925 XMX1 = 1.207 XMX2 = 1.476

ALPH1 = 61.130 ALPH2 = 52.603 PT2/PT0 = 0.4847 PT0/P2 = 31.502 T2/T0 = 0.4585 V/VCR2 = 1.802 EBAR = 0.13064 ETA-V = 0.8034

ITERATION COMPLETED

BETA(JUT) = -61.1300 DEG

G*(IN) = 0.59435

G*(JUT) + DEL(FINAL) = 0.59712

G*(IN) = 0.5944

SIGMA(IN) = 2.7603

G*(JUT) = 0.4210

C* = 1.5405

SIGMA(OUT) = 3.966

01 UNIT05, EOF.

REC= 00000 FIL= 00002

(3) Output 3 lists the properties calculated for the upstream conditions:

PSZ	upstream static pressure, N/m^2 ; lbf/ft^2
TSZ	upstream static temperature, K; $^{\circ}\text{R}$
UZ	upstream velocity, m/sec ; ft/sec
ASZ (ATZ)	speed of sound based on upstream static (total) temperature, m/sec ; ft/sec
RHSZ (RHTZ)	static (total) density based on upstream static (total) temperature, kg/m^3 ; slug/ft^3
MUSZ (MUTZ)	dynamic viscosity based on upstream static (total) temperature, $(\text{N})(\text{sec})/\text{m}^2$; $(\text{lbf})(\text{sec})/\text{ft}^2$
NUSZ (NUTZ)	kinematic viscosity based on upstream static (total) temperature, m^2/sec ; ft^2/sec
CP	specific heat at constant pressure, $\text{J}/(\text{kg})(\text{K})$; $(\text{ft})(\text{lbf})/(\text{slug})(^{\circ}\text{R})$
PR	Prandtl number
TC	thermal conductivity, $\text{J}/(\text{m})(\text{sec})(\text{K})$; $(\text{ft})(\text{lbf})/(\text{ft})(\text{sec})(^{\circ}\text{R})$
ARCL	total distance along surface, m ; ft

The next part of this output gives the variables describing the flow along the surface:

PRES	static pressure, N/m^2 ; lbf/ft^2
UE	free-stream velocity, m/sec ; ft/sec
ME	free-stream Mach number
POPTZ	static-to-total pressure ratio
VOVCR	critical velocity ratio

(4) Output 4 corresponds to KMAIN. It indicates the regions of laminar and turbulent boundary layers, and the stations at which instability, transition, and separation occur. It gives all the principal boundary-layer output parameters:

X	X-coordinate, m ; ft
S	surface length, m ; ft
DELSR	displacement thickness, m ; ft
THET	momentum thickness, m ; ft

DELTA boundary-layer thickness, m; ft
 FORM compressible form factor
 FORMI incompressible form factor

The next part of the output gives the skin-friction and heat-transfer parameters:

CF skin friction coefficient at wall
 TAUW shear stress at wall, N/m^2 ; lbf/ft²
 RTH momentum-thickness Reynolds number
 DTDY slope of temperature profile at wall, K/m; °R/ft
 NUSS local Nusselt number
 HTRAN heat transfer per unit area, $J/(sec)(m^2)$; (ft)(lbf)/(sec)(ft²)
 CRN Reynolds analogy parameter

(5) Output 5 gives the ideal and corrected surface coordinates. Translated surface coordinates are also provided so that coordinates for a blade profile may be obtained.

(6) After the output for both surfaces is given, the final output, numbered 6, lists the aftermixing properties. Variable names ending in 0 refer to the upstream station, while those ending in 1 refer to the rotor exit (before mixing) and those ending in 2 refer to the mixed relative conditions downstream of the rotor. The output consists of rotor exit free-stream Mach number (XMFS1), rotor exit spacing (SPACING) in m (ft), trailing-edge thickness (TE) in m (ft), Mach number (XM2), axial Mach numbers (XM1 and XM2), critical velocity ratios (V/VCR1 and V/VCR2), flow angles measured from axial direction (ALPH1 and ALPH2) in degrees, total-to-total pressure ratio (PT2/PT0), total-to-static pressure ratio (PT0/P2), static-to-total temperature ratio (T2/TT0), rotor kinetic energy loss coefficient (EBAR2), and rotor efficiency (ETA-N). Also printed are some miscellaneous parameters for the corrected blade profile including the final outlet flow angle and solidity.

PROGRAM DESCRIPTION

The program SSROTR designs supersonic rotor blades corrected for boundary-layer displacement thickness. The program uses the input value of the outlet flow angle (which controls the outlet circular turning) for the initial design. If the corrected outlet spacing is not equal to the inlet spacing, the program estimates a new value of outlet flow angle and repeats the design. The iteration procedure is essentially a half-interval search. The program input consists essentially of the surface Mach number distribution

and total flow conditions. The primary output is the corrected blade profile and the boundary-layer parameters.

The program SSROTR is logically divided into two parts: (1) an ideal rotor design section, and (2) a section to calculate the boundary-layer parameters and aftermixing conditions. The ideal rotor design section consists principally of subroutines ROTORU and ROTORR. The boundary-layer section is controlled by subroutine BLAYR. The major subroutines called by BLAYR are PRECAL, LAMNAR, TURBLN, and AFMIX.

SUBROUTINE ROTORU

Subroutine ROTORU calculates the upper- and lower-surface transition arcs by the method of characteristics. The description of the method and the main dictionary of variables are given in reference 2. This routine is called only once for any given blade design. The program variables for ROTORU that are not in reference 2 are

ETALOW	angle of tangent to lower surface, measured from axial direction
ETAUP	angle of tangent to upper surface, measured from axial direction
GAMMI	$GAM-1, \gamma - 1$
GAMPI	$GAM+1, \gamma + 1$
STML	lower-surface critical velocity ratio
STMU	upper-surface critical velocity ratio

SUBROUTINE ROTORR

Subroutine ROTORR calculates the rotated ideal passage coordinates, as well as the inlet and outlet passage spacing based on the current value of outlet flow angle. This routine is called once for each iteration of outlet flow angle. The description of the method and the main dictionary of variables are given in reference 2. The variables for ROTORR that are not in reference 2 are

ETACL	angle of flow along lower-surface circular arc
ETACU	angle of flow along upper-surface circular arc
ETALN	angle of flow along lower-surface inlet transition arc
ETALO	angle of flow along lower-surface outlet transition arc
ETASN	absolute value of inlet flow angle

ETASO	absolute value of outlet flow angle
ETAUN	angle of flow along upper-surface inlet transition arc
ETAUO	angle of flow along upper-surface outlet transition arc
SMLN	lower-surface critical velocity ratio for inlet
SMLO	lower-surface critical velocity ratio for outlet
SMUN	upper-surface critical velocity ratio for inlet
SMUO	upper-surface critical velocity ratio for outlet

SUBROUTINE BLAYR

Subroutine BLAYR provides calls for the calculation of the boundary-layer characteristics and the mixing losses. The major subroutines called by BLAYR are PRECAL, LAMNAR, TURBLN, and AFMIX. These, in turn, call several other subroutines. The methods used for calculating the boundary-layer characteristics and the program description are given in reference 4. The Lagrangian interpolation routine of reference 4 has been replaced by a linear interpolation.

Since a highly favorable pressure gradient exists for some portions of the blade surfaces, the range of the equations of reference 6 were extended by the method given in reference 6. These changes were made in LAMNAR and are for SW (temperature function at the wall) = 0. These changes are noted in the program with comment cards. Two curve-fit ranges were also extended. $RCRIT$ and $DIFF$ were extended as follows: $RCRIT = 8.3163$ when $SHAPK$ is greater than 0.07; and $DIFF = 44\ 000\ KBAR + 700$ when $KBAR$ is greater than 0.03.

SUBROUTINE AFMIX

Subroutine AFMIX takes the boundary-layer parameters (displacement and momentum thicknesses) and the free-stream conditions at the rotor exit and calculates the aftermixing conditions by the method described in reference 5. In this loss model the flow sufficiently downstream of the blades is assumed to be mixed to uniform conditions.

The calling sequence for AFMIX is `CALL AFMIX (ALPHI, DELS, DELP, THETS, THETP, TE, SP, XMFSI)`, where

ALPHI	rotor exit flow angle measured from axial direction
DELS	displacement thickness at rotor exit for suction (upper) surface
DELP	displacement thickness at rotor exit for pressure (lower) surface

THETS momentum thickness at rotor exit for suction surface
 THETP momentum thickness at rotor exit for pressure surface
 TE trailing-edge thickness
 SP exit spacing
 XMFSI free-stream Mach number at rotor exit

The subroutine description and the main dictionary of variables are given in reference 3.

PROGRAM LISTING

```
$IIFTC SSROTR DECK

C    COMBINED ROTOR-BOUNDARY LAYER PROGRAM

      COMMON/COE/KOE,DELSRL,DELSRU
      COMMON/CTOBL/ALPH1,SPA,XMAX,TE,NTURBU,NTURPL,CTHETU,CTHETL
      COMMON/LNK1/XOML(100),YOML(100),SML(100),ETAML(100),NL,XOMU(100),
1    YOMU(100),SMU(100),ETAMU(100),NU
      COMMON/LNK2/SMIN,SMOUT,BETAT,CSTAR,SIGMAO,GSTAP,GSTARI,GAMMA,
*    CONVER,RECONV
      COMMON/ROT1/V(4),BETAN,NKJ(5),PR(4)
      COMMON/C1/DUM1(7),NTURB,DUM2(6),CTHET,DUM3(910)

      DIMENSION X(100),Y(100)

1    BETAP = 0.
      BETAM = 0.
C    CALCULATE UNROTATED BLADE COORDINATES
      CALL ROTORU
      CALL INPUT1
      CALB = 0.
C    CALCULATE ROTATED BLADE COORDINATES
2    CALL RCTORR (CALB)
      CALB = 1.
C    NORMALIZE ROTATED BLADE COORDINATES TO CHORD LENGTH
      DO 3 I=1,NU
        X(I) = XOMU(I)/CSTAR
3    Y(I) = YOMU(I)/CSTAR
      KOE = 0
      NTURB = NTURBU
      CTHET = CTHETU
      CALL INPUT2 (X,Y,SMU,ETAMU,NU)
C    CALCULATE UPPER SURFACE BOUNDARY LAYER
      CALL PLAYP
C    NORMALIZE ROTATED BLADE COORDINATES TO CHORD LENGTH
      DO 4 I=1,NL
        X(I) = XOML(I)/CSTAR
```

```

4 Y(I) = YCML(I)/CSTAR
  KODE = 1
  NTURB = NTURBL
  CTHET = CTHETL
  CALL INPUT2 (X,Y,SML,ETAML,NL)
C   CALCULATE LOWER SURFACE BOUNDARY LAYER
  CALL ELAYR

C   CHECK FOR EQUAL SPACING AT INLET AND OUTLET
  DELTOT = (DELSRL + DELSRU)*CSTAR
  OUTSPA = GSTAPO + DELTOT/COS(BETAT*CONVER)
  ELIMIT = .0001*CSTAR/XMAX
  IF (ABS(GSTARI - OUTSPA) .LE. ELIMIT) GO TO 5
  IF (OUTSPA .GE. GSTARI) BETAP = BETAT
  IF (OUTSPA .LT. GSTARI) BETAM = BETAT
  GSTARO = OUTSPA
  IF (BETAP .NE. 0. .AND. BETAM .NE. 0.) GO TO 6
C   CALCULATE NEW OUTLET FLOW ANGLE
  BETAT = -ARCOS(COS(BETAN)*SMIN/SMOUT*((1. + (GAMMA-1.)/2.*
* SMOUT*SMOUT)/(1. + (GAMMA-1.)/2.*SMIN*SMIN))**((GAMMA+1.)/(2.*
* GAMMA-1.))) + DELTOT/GSTARI)*PFCONV
  GO TO 2
6 BETAT = (BETAP + BETAM)/2.
  GO TO 2

5 WRITE (6,100) BETAT,GSTARI,OUTSPA
100 FORMAT (//5X,19HITERATION COMPLETED/7X,11HBETA(OUT) =,F10.4,4H DEG
*,10X,8HG*(IN) =,F10.5,10X,22HG*(OUT) + DEL(TOTAL) =,F10.5)
  CALB = 2.
  CALL ROTORR (CALB)
  GO TO 1

END
$IEFTC INPU1 DECK
  SUBROUTINE INPUT1

C   BOUNDARY LAYER INPUT

  COMMON/LTU/LDAT
  COMMON/LNK2/UPMAC,SMOUT,ALP1M,CSTAR,SIGMAO,GSTARO,GSTARI,GAMMA,
* CONVER,RECONV
  COMMON/CTOBL/ALPH1,SPA,XMAX,TE,NTUPRU,NTUPRL,CTHETU,CTHETL
  COMMON/C1/GAM,R,PTZ,TTZ,UPMACH,NST,NVP,NTURB,KPVM,KEM,KSMTH,
1KSPLN,KLE,KATCH,CTHFT,DIAM,TLAM,DTUDE,TTURB,KPRE,KGRAD,KSDE,KLAM,
2KMAIN,KPROF,X(100),Y(100),PRFS(100),UF(100),ME(100),PORTZ(100),
3VOVCR(100),TWAL(100),ETA(100)

  READ (LDAT,100) NTURBU,NTURBL,NVP,F,PTZ,TTZ,XMAX,TE,CTHETU,CTHETL
100 FORMAT (3I3,1X,7(F9.5,1X))
  READ (LDAT,1010) KPRE,KGRAD,KSDE,KLAM,KMAIN,KPROF,KEM
1010 FORMAT (7I5)

  GAM = GAMMA
  UPMACH = UPMAC

  RETURN
END

```

\$IEFTC ROTU LIST,DECK
SUBROUTINE ROTORU

C****CALCULATE THE UNROTATED BLADE COORDINATES

```
COMMON/LTU/LDAT
COMMON/FACTOR/PERM,GAMM1,GAMP1,GAM
COMMON/ROT1/VIN,VOUT,VLOW,VUP,BETAN,NPER,KMAXN,KMAXO,JMAXN,JMAXO,
1 RIN,ROUT,RLOW,RUP
COMMON/ROT2/XLOW(400),YLOW(400),STEL(400),ETALOW(400),KNDEX,
1 XUP(400),YUP(400),STMU(400),ETAUP(400),JNDEX
COMMON/LNK2/SMIN,SMOUT,BETAT,CSTAR,SIGMAO,GSTARO,GSTARI,GAMMA,
* CONVER,RECONV
```

LOGICAL ANGLE

EXTERNAL PCFRS

DATA LDAT/5/

$F(V, FN) = 2 * V - HALFPI * (PERM - 1.) - 2 * (FN - 1.) * DELV$

```
1 READ (LDAT,11) BETAN,VIN,VLOW,VUP,VOUT,BETAT,DELV,GAM
11 FORMAT (8(F6.2,2X))
```

CC CONVERSION FACTORS AND CONSTANTS

```
GAMMA = GAM
CONVER = .174532925E-01
RECONV = 57.2957796
HALFPI = 3.14159265/2.
```

C ONE POINT WILL BE PRINTED FOR EVERY NPER POINTS CALCULATED

```
NPER = 10
IF (DELV .GE. 0.2) NPER = 1
GAMP1 = (GAM + 1.) / 2.
GAMM1 = (GAM - 1.) / 2.
PERM = SQRT(GAMP1/GAMM1)
X0 = 1./PERM
X2 = 0.999999999
XINTL = (X0 + X2) / 2.
```

```
ANGLE = .TRUE.
IF (VLOW .LE. AMIN1(VUP,VIN,VOUT)) GO TO 120
WRITE (6,119)
```

```
119 FORMAT (/31X,70HV(LOW) MUST BE LESS THAN OR EQUAL TO THE MINIMUM
1OF V(UP),V(IN),V(OUT))
ANGLE = .FALSE.
```

```
120 IF (VUP .GE. AMAX1(VIN,VOUT)) GO TO 118
WRITE (6,117)
```

```
117 FORMAT (/33X,66HV(UP) MUST BE GREATER THAN OR EQUAL TO THE MAXIMU
1M OF V(IN),V(OUT))
ANGLE = .FALSE.
```

```
118 VUMAX = HALFPI * (PERM - 1.) * RECONV
IF (VUP .LE. VUMAX) GO TO 116
WRITE (6,115) VUMAX
```

```
115 FORMAT (/41X,37HV(UP) MUST BE LESS THAN V(UP) (MAX) = ,F9.4,4H DEG
1)
ANGLE = .FALSE.
```

```

116 IF (.NOT. ANGLE) GO TO 1

WRITE (6,97)
97 FORMAT (1H1,57X,17HDESIGN PARAMETERS)

CC MISCELLANEOUS CALCULATIONS
DELV = DELV*CONVER
FN = 1.
V = VIN*CONVER
DO 4 I=1,2
POFX = F(V,FN)
CALL FOOT (X0,X2,XINTL,FOFX,FOFRS,X1)
IF (I .EQ. 2) GO TO 4
RIN = X1
V = VOUT*CONVER
4 CONTINUE
ROUT = X1

SSMIN = 1./RIN
SSMOUT = 1./ROUT

SMS = SSMIN
I = 1
16 SM = SQRT(((1./GAMP1)*SMS*SMS)/(1.-(GAMM1/GAMP1)*SMS*SMS))
GO TO (17,18,19,20),I
17 SMIN = SM
SMS = SSMOUT
I = 2
GO TO 16
18 SMOUT = SM
DELV = DELV*RECONV

CC PRINT ALL DESIGN PARAMETERS
WRITE (6,95) BETAN,VIN,VUP,VOUT,PFTAT
95 FORMAT (/2X,10HBETA(IN) =,F8.4,4H DEG,4X,7HV(IN) =,F8.4,4H DEG,6X,
1 7HV(UP) =,F9.4,4H DEG,7X,8HV(OUT) =,F8.4,4H DEG,4X,11HBETA(OUT) =
2,F9.4,4H DEG)
WRITE (6,94) DELV, VLOW, GAM
94 FORMAT (/20X,9HDELTA V =,F8.4,4H DEG,11X,8HV(LOW) =,F8.4,4H DEG,
1 11X,7HGAMMA =,F8.4)

CC CONVERT FROM DEGREES TO RADIANS
VIN = VIN*CCNVER
VOUT = VOUT*CONVER
VUP = VUP*CCNVER
VLOW = VLOW*CONVER
BETAN = BETAN*CCNVER
DFLV = DFLV*CCNVER

CC CHOOSE LONGEST TRANSITION ARC OF LOWER SURFACE
VNL = VIN - VLOW
KMAXN = (VNL/DELV) + 0.5
VCL = VOUT - VLOW
KMAXO = (VCL/DELV) + 0.5
KMN = MAXO(KMAXN,KMAXO)
V = AMAX1(VIN,VOUT)

```

```

CC  CALCULATE  R*(LOW)=RLOW,  M*(LOW)=SSMLCW,  M(LOW)=SMLOW
      IF (VLOW .EQ. 0.0) GO TO 2
      FN = KMN + 1
      FOPX = F(V,FN)
      CALL ROOT (X0,X2,XINTL,FCFX,FOFRS,RLOW)
      GO TO 3
2  RLOW = 1.0
3  SSMLCW = 1./RLOW
      SMS = SSMLCW
      I = 3
      GO TO 16
19  SMLOW = SM

CC  SET INITIAL POINTS FOR LOWER ARC CALCULATIONS
      KINDEX = KMN/NPER
      KDEX = KINDEX
      STML(KDEX+1) = SSMLCW
      XLOW(KDEX+1) = 0.0
      YLOW(KDEX+1) = RLOW
      ETALOW(KDEX+1) = 0.
      PHIKP1 = -(V-VLOW) + FLOAT(KMN)*DELV
      UMKP1 = ARSIN(SQRT(GAMP1*RLOW*RLOW - GAMM1))
      TXLO = XLOW(KDEX+1)
      TYLO = YLOW(KDEX+1)

CC  CHOOSE LONGEST TRANSITION ARC OF UPPER SURFACE
      VUT = VUP - VOUT
      JMAXO = (VUT/DELV)+0.5
      VUI = VUP - VIN
      JMAXN = (VUI/DELV)+0.5
      JMN = MAX0(JMAXO,JMAXN)
      V = AMIN1(VOUT,VIN)

CC  CALCULATE  R*(UP)=RUP,  M*(UP)=SSMUP,  M(UP)=MUP
      FN = -(JMN+1) + 2
      FOPX = F(V,FN)
      CALL ROOT (X0,X2,XINTL,FCFX,FOFRS,RUP)
      SSMUP = 1./RUP
      SMS = SSMUP
      I = 4
      GO TO 16
20  SMUP = SM

CC  SET INITIAL POINTS FOR UPPER ARC CALCULATIONS
      JINDEX = JMN/NPER
      JDEX = JINDEX
      STMU(JDEX+1) = SSMUP
      XUP(JDEX+1) = 0.0
      YUP(JDEX+1) = RUP
      ETAUP(JDEX+1) = 0.
      PHIJP1 = -(VUP-V) + FLOAT(JMN)*DELV
      UMJP1 = ARSIN(SQRT(GAMP1*RUP*RUP - GAMM1))
      TXUP = XUP(JDEX+1)
      TYUP = YUP(JDEX+1)

      IF (VIN .EQ. VLOW .AND. VLOW .EQ. VOUT) GO TO 100

C****CALCULATE COORDINATES FOR LOWER TRANSITION ARC - UNROTATED

```



```

KDEX = KINDEX + 1
NUM = 0
V = AMAX1(VIN,VOUT)
KEEP = KMN + 1
DO 30 KK=1,KMN
K = KEEP - KK
NUM = NUM + 1
PHIK = PHIKF1 - DELV
FN = K
FOFX = F(V,FN)
CALL ROOT (X0,X2,XINTL,FOFX,FOFRS,TR)
TX = TR*SIN(PHIK)
TY = TR*COS(PHIK)
EMWK = TAN(-PHIKP1)
UMK = ARSIN(SQRT(GAMP1*TR*TR - GAMM1))
EMK = -TAN((PHIK+UMK+PHIKP1+UMKP1)/2.)
TEMP = TYLO - EMWK*TXLO
TEMPP = TY - EMK*TX
TEMPPP = EMK - EMWK
TXLO = (TEMP - TEMPP)/TEMPPP
TYLO = ((EMK*TEMP) - (EMWK*TEMPP))/TEMPPP
PHIKP1 = PHIK
UMKP1 = UMK

```

CC SAVE EVERY "NPER-TH" POINT

```

N = NUM - (NUM/NPER)*NPER
IF (N .GT. 0) GO TO 30
KDEX = KDEX - 1
STML(KDEX) = 1./TR
XLOW(KDEX) = TXLO
YLOW(KDEX) = TYLO
ETALOW(KDEX) = -PHIK

```

30 CONTINUE

100 IF (VIN .EQ. VUP .AND. VUP .EQ. VOUT) GO TO 200

C*****CALCULATE COORDINATES FOR UPPER TRANSITION ARC - UNROTATED

```

JDEX = JINDEX + 1
NUM = 0
V = AMIN1(VOUT,VIN)
JEEP = JMN + 1
DO 41 JJ=1,JMN
J = JEEP - JJ
NUM = NUM + 1
PHIJ = PHIJF1 - DELV
FN = -J + 2
FOFX = F(V,FN)
CALL ROOT (X0,X2,XINTL,FOFX,FOFRS,TR)
TX = TR*SIN(PHIJ)
TY = TR*COS(PHIJ)
EMWJ = TAN(-PHIJP1)
UMJ = ARSIN(SQRT(GAMP1*TR*TR - GAMM1))
EMJ = TAN((-PHIJ+UMJ-PHIJP1+UMJP1)/2.)
TEMP = TYUP - EMWJ*TXUP
TEMPP = TY - EMJ*TX
TEMPPP = EMJ - EMWJ
TXUP = (TEMP - TEMPP)/TEMPPP
TYUP = ((EMJ*TEMP) - (EMWJ*TEMPP))/TEMPPP

```

```

      PHIJP1 = PHIJ
      UMJP1 = UMJ

CC      SAVE EVERY "NPER-TH" PCINT
      N = NUM - (NUM/NPER)*NPER
      IF (N .GT. 0) GO TO 41
      JDEX = JDEX - 1
      STMU(JDEX) = 1./TR
      XUP(JDEX) = TXUP
      YUP(JDEX) = TYUP
      ETAUP(JDEX) = -PHIJ

      41 CONTINUE

CC      MISCELLANEOUS OUTPUT
      200 WRITE (6,622)
      622 FORMAT (//54X,24HMISCELLANEOUS PARAMETERS//)
      WRITE (6,1000) SSMIN,SMIN,SMOUT,SSMOUT
      1000 FORMAT (/25X,8HM*(IN) =,F9.4,3X,7HM(IN) =,F9.4,10X,8HM(OUT) =,F9.4
      1,5X,9HM*(OUT) =,F9.4)
      WRITE (6,1001) SSMLOW,SMLOW,SMUP,SSMUP
      1001 FORMAT (/25X,9HM*(LOW) =,F9.4,2X,8HM(LOW) =,F9.4,11X,7HM(UP) =,
      * F9.4,6X,8HM*(UP) =,F9.4)

      KNDEX = KNDEX + 1
      JNDEX = JNDEX + 1

      RETURN
      END
$IEFTC ROO      DECK
      SUBROUTINE ROOT (X0,X2,XINTL,FOFX,FUNC,X1)

      DOUBLE PRECISION X,XX0,XX2

C      WE ARE SEEKING AN X SUCH THAT FUNC(X) = FOFX WHERE FOFX IS A KNOWN
C      FUNCTIONAL VALUE
C      1 LOCATE FOFX IN (F0,FX) OR (FX,F2) WHERE FX IS THE PREVIOUS
C      APPROXIMATION TO FOFX
C      2 LET X = 1/2(XX0+X) OR X = 1/2(X+XX2)
C      3 IS FUNC(X) = FOFX ? IF NOT, REPEAT PROCEDURE

      XX0 = X0
      XX2 = X2
      F0 = FUNC(XX0)
      F2 = FUNC(XX2)
      IF ( FOFX .LT. F0 .AND. FOFX .LT. F2 .OR. FOFX .GT. F0 .AND.
      1FOFX .GT. F2 ) GO TO 1005

      IF (ABS(FOFX-F0) .LE. .00001) GO TO 1007
      IF (ABS(FOFX-F2) .LE. .00001) GO TO 1008

      X = XINTL
      KOUNT = 0
      1000 X1 = X
      KOUNT = KOUNT + 1
      A = FOFX - F2
      FX = FUNC(X)
      IF (KOUNT .GE. 60) WRITE (6,1004) KOUNT,X,FX,FOFX

```

```

1004 FORMAT (14L,9H KOUNT ,G16.9,9H X ,G16.9,9H FX ,G16.9,
19H FOFX ,G16.9)
IF (ABS(FX-FOFX) .LE. .00001) RETURN
IF (KOUNT .EQ. 75) GO TO 1002
IF (A*(FX-FCFX) .LT. 0.) GO TO 1001
XX0 = X
X = (X+XX2)/2.
GO TO 1000
1001 XX2 = X
X = (XX0+X)/2.
F2 = FX
GO TO 1000

1002 WRITE (6,1003)
1003 FORMAT (//30X,62H75 ITERATIONS HAVE BEEN PERFORMED WITHOUT CONVERG
ING TO A ROOT)
RETURN

1005 WRITE (6,1006) FOFX
1006 FORMAT (//10X,7HF(X) = ,G16.9,31H IS OUTSIDE OF SPECIFIED LIMITS)
RETURN

1007 X1 = X0
RETURN

1008 X1 = X2
RETURN

END
$IBFTC FELI DECK
FUNCTION FOFRS (X)

DOUBLE PRECISION X
COMMON/FACTOR/PERM,GAMM1,GAMP1,GAM

ARG1 = 2.*GAMM1/(X*X) - GAM
ARG2 = 2.*GAMP1*X*X - GAM
IF (ABS(ARG1) .GT. 1.0 .OR. ABS(ARG2) .GT. 1.0) WRITE (6,1) ARG1
1,ARG2
1 FORMAT (//14X,61HARGUMENT OF ARCSIN IS OUTSIDE DOMAIN OF DEFINITIO
1N ARG1 = ,G16.9,11H ARG2 = ,G16.9)

FOFRS = PERM*ARSIN(ARG1) + APSIN(APG2)

RETURN
END
$IBFTC ROTR LIST,DECK
SUBROUTINE ROTORS (CALB)

C CALCULATE (1) ROTATED BLADE COORDINATES
C (2) INLET AND OUTLET ELAPE SPACING

DIMENSION XLOWN(80),YLOWN(80),SMLN(80),ETALN(80),XCLOW(35),
1 YCLOW(35),ETACL(35),XLOWO(80),YLOWO(80),SMLO(80),ETALO(80)
DIMENSION XSN(11),YSN(11),ETASN(11),XUPN(80),YUPN(80),SMUN(80),
1 ETAUN(80),XCUP(35),YCUP(35),ETACU(35),XUPO(80),YUPO(80),SMUO(80),
2 ETAUO(80),XSO(11),YSO(11),FTASO(11)

COMMON/LNK1/XOML(100),YOML(100),SML(100),ETAML(100),NL,XOMU(100),

```

```

1 YOMU(100),SMU(100),ETAMU(100),NU
COMMON/POT1/VIN,VOUT,VLOW,VUP,BETAN,NPER,KMAXN,FMAXO,JMAXN,JMAXO,
1 RIN,ROUT,RLCW,RUP
COMMON/ROT2/XLOW(400),YLOW(400),STML(400),ETALOW(400),KNDEX,
1 XUP(400),YUP(400),STMU(400),ETAUP(400),JINDEX
COMMON/LNK2/SMIN,SMOUT,BETAT,CSTAR,STGMAO,GSTARO,GSTARI,GAMMA,
* CONVER,FECCNV

IF (CALB .NE. 0.) GO TO 2
WRITE (6,100) BETAT
100 FORMAT (//10X,15HINPUT BETAT(0) =,F10.4,4H DPG//)
GO TO 3

2 IF (CALB .NE. 1.) GO TO 3
WRITE (6,101) BETAT,GSTARO
101 FORMAT (//10X,20HCALCULATED BETAT(0) =,F10.4,4H DPG,5X,14HWHEN G*(O
1UT) =,F10.5)

3 BETAT = BETAT*CONVER

ALPHLN = (VIN - VLOW) - BETAN
ALPHLO = -(VOUT - VLOW) - BETAT
IF (ALPHLN .LE. 0. .AND. ALPHLO .GE. 0.) GO TO 4
WRITE (6,102)
102 FORMAT (//27X,79HV(LOW) MUST BE GREATER THAN OR EQUAL TO V(IN) - B
1ETA(IN) AND V(OUT) + BETAT(OUT))
STOP

C****CALCULATE COORDINATES FOR LOWER TRANSITION ARC - ROTATED
4 SINAIN = SIN(ALPHLN)
COSALN = COS(ALPHLN)
ABSALN = ABS(ALPHLN)
SINALO = SIN(ALPHLO)
COSALO = COS(ALPHLO)
ABSALO = ABS(ALPHLO)
KDEX = KNDEX + 1
KN = (KMAXN/NPER) + 2
KO = (KMAXO/NPER) + 2

DO 5 KK=1,KNDFX
K = KDEX - KK
KN = KN - 1
KO = KO - 1
IF (KN .IE. 0) GO TO 6
SMLN(KN) = STML(K)
XLOWN(KN) = YLOW(K)*SINAIN + XLOW(K)*COSALN
YLOWN(KN) = YLOW(K)*COSALN - XLOW(K)*SINALN
ETALN(KN) = ETALOW(K) + ABSALN
6 IF (KO .IE. 0) GO TO 5
SMLO(KO) = STML(K)
XLOWC(KO) = YLOW(K)*SINALO - XLOW(K)*COSALO
YLOWC(KO) = YLOW(K)*COSALO + XLOW(K)*SINALO
ETALC(KO) = ETALOW(K) + ABSALO
5 CONTINUE

ALPHUN = (VUP - VIN) - BETAN
ALPHUC = -(VUP - VOUT) - BETAT
IF (ALPHUN .LE. 0. .AND. ALPHUC .GE. 0.) GO TO 7
WRITE (6,103)

```

```

103 FORMAT (//2X,75HV(UF) MUST BE LESS THAN OR EQUAL TO V(IN) + BETA(I
1N) AND V(OUT) - BETA(OUT))
STOP

```

```

C****CALCULATE COORDINATES FOR UPPER TRANSITION ARC - ROTATED

```

```

7 SINAUN = SIN(ALPHUN)
  COSAUN = COS(ALPHUN)
  ABSAUN = ABS(ALPHUN)
  SINAUO = SIN(ALPHUO)
  COSAUO = COS(ALPHUO)
  ABSAUO = ABS(ALPHUO)
  JDEX = JNDEX + 1
  JN = (JMAXN/NPER) + 2
  JO = (JMAXO/NPER) + 2

  DO 8 JJ=1,JNDEX
  J = JDEX - JJ
  JO = JO - 1
  JN = JN - 1
  IF (JO .LE. 0) GO TO 9
  SMUO(JO) = STMU(J)
  XUPO(JO) = YUP(J)*SINAUO - XUP(J)*COSAUO
  YUPO(JO) = YUP(J)*COSAUC + XUP(J)*SINAUO
  ETAUO(JO) = ETAUP(J) + ABSAUO
9 IF (JN .LE. 0) GO TO 8
  SMUN(JN) = STMU(J)
  XUPN(JN) = YUP(J)*SINAUN + XUP(J)*COSAUN
  YUPN(JN) = YUP(J)*COSAUN - XUP(J)*SINAUN
  ETAUN(JN) = ETAUP(J) + ABSAUN
8 CONTINUE

```

```

C****CALCULATE THE INLET AND OUTLET (DIMENSIONLESS) BLADE SPACING

```

```

  YLASTI = YUPN(1) + TAN(BFTAN)*(XLOWN(1) - XUPN(1))
  GSTARI = YLOWN(1) - YLASTI
  YLASTO = YUPO(1) + TAN(BETAT)*(XLOWO(1) - XUPO(1))
  GSTARO = YLOWO(1) - YLASTO

```

```

  DALPH = 5.*CONVER

```

```

C****CIRCULAR ARC (LOWER)

```

```

  ALPH = ALPHLO + DALPH
  ALPLOW = ALPHLN
  KOUNT = 0
10 KOUNT = KOUNT + 1
  XCLW(KOUNT) = FLOW*SIN(ALPLOW)
  YCLW(KOUNT) = FLOW*COS(ALPLOW)
  ETACL(KOUNT) = ABS(ALPLOW)
  ALPLOW = ALPLOW + DALPH
  IF (ABS(ALPH-ALPLOW) .LE. .001) GO TO 11
  IF (ALPHIO .LT. ALPLOW .AND. ALPLOW .LT. ALPH) ALPLOW = ALPHLO
  GO TO 10
11 NCL = KOUNT

```

```

C****CIRCULAR ARC (UPPER)

```

```

  ALPH = ALPHUO + DALPH
  ALPHUP = ALPHUN
  KOUNT = 0

```

```

12 KOUNT = KOUNT + 1
   XCUP(KOUNT) = RUP*SIN(ALPHUP)
   YCUP(KOUNT) = RUP*COS(ALPHUP)
   ETACU(KOUNT) = ABS(ALPHUP)
   ALPHUP = ALPHUP + DALPH
   IF (ABS(ALPH-ALPHUP) .LE. .001) GO TO 13
   IF (ALPHUO .LT. ALPHUP .AND. ALPHUP .LT. ALPH) ALPHUP = ALPHUO
   GO TO 12
13 NCU = KOUNT

```

C****CALCULATE COORDINATES FOR STRAIGHT LINE POPTION OF UPPER ABC

```

   KOUNT = 1
   IF (XLOWN(1) .LE. XUPN(1)) GO TO 15
   WRITE (6,106)
106 FORMAT (8H0(ROTR),4X,71HUPPER SURFACE INLET LONGER THAN LOWER SUR
   *FACE INLET - CASE TERMINATED)
   STOP
15 DELXI = (XUPN(1) - XLOWN(1))/10.
   IF (XLOWO(1) .GE. XUPO(1)) GO TO 16
   WRITE (6,105)
105 FORMAT (8H0(ROTR),4X,73HUPPER SURFACE OUTLET LONGER THAN LOWER SU
   *FACE OUTLET - CASE TERMINATED)
   STOP
16 DELXO = (XLOWO(1) - XUPO(1))/10.
   XSIN = XUPN(1)
   YSIN = YUPN(1)
   XSOUT = XUPC(1)
   YSOUT = YUPO(1)
   XSN(KOUNT) = XSIN
   YSN(KOUNT) = YSIN
   ETASN(KOUNT) = ETAUN(1)
   XSO(KOUNT) = XSOUT
   YSO(KOUNT) = YSOUT
   ETASO(KOUNT) = ETAUO(1)
   TANBN = TAN(ETAN)
   ABSBN = ABS(ETAN)
   TANBO = TAN(ETAT)
   ABSBT = ABS(ETAT)
   DO 14 I=2,11
   XSIN = XSIN - DELXI
   XSOUT = XSOUT + DELXO
   YSIN = YUPN(1) + TANBN*(XSIN - XUPN(1))
   YSOUT = YUPO(1) + TANBO*(XSOUT - XUPO(1))
   ETASN(I) = ABSBN
   XSN(I) = XSIN
   YSN(I) = YSIN
   ETASO(I) = ABSBT
   XSO(I) = XSOUT
14 YSO(I) = YSOUT

```

C****PREPARE DATA FOR BOUNDARY LAYER PROGRAM

```

   CSTAR = SQRT((XLOWO(1) - XLOWN(1))**2 + (YLOWO(1) - YLOWN(1))**2)
   SIGMAI = CSTAR/GSTAPI
   SIGMAO = CSTAR/GSTAFO
   WRITE (6,104) GSTARI,SIGMAI,CSTAR,GSTAFO,SIGMAO

```

```

104 FORMAT (//39X,8HG*(IN) =,F9.4,37X,11HSIGMA(IN) =,F9.4/69X,4HC* =,
* F9.4/39X,9HG*(OUT) =,F9.4,36X,12HSIGMA(OUT) =,F9.4//)
BETAT = BETAT*RECONV
SSMLCW = 1./RLOW
SSMUP = 1./RUP
SSMIN = 1./RIN
SSMOUT = 1./ROUT
IF (CALB.EQ. 2.) RETURN

```

C STORE ROTATED BLADE COORDINATES FOR LOWER SURFACE

```

KN = (KMAXN/NPER) + 1
DO 310 I=1,KN
XOML(I) = XLOWN(I)
YOML(I) = YLOWN(I)
SML(I) = SMLN(I)
ETAML(I) = ETALN(I)
310 CONTINUE
NL = KN
NCL = NCL-1
DO 311 I=2,NCL
NL = NL + 1
XOML(NL) = XCLOW(I)
YOML(NL) = YCLOW(I)
SML(NL) = SSMLOW
ETAML(NL) = ETACL(I)
311 CONTINUE
KO = (KMAXO/NPER) + 1
DO 312 I=1,KO
J = KO + 1 - I
NL = NL + 1
XOML(NL) = XLOWO(J)
YOML(NL) = YLOWO(J)
SML(NL) = SMLO(J)
ETAML(NL) = ETALO(J)
312 CONTINUE

```

C STORE ROTATED BLADE COORDINATES FOR UPPER SURFACE

```

DO 313 I=1,10
J = 12 - I
XOMU(I) = XSN(J)
YOMU(I) = YSN(J)
SMU(I) = SSMIN
ETAMU(I) = ETASN(J)
313 CONTINUE
NU = 10
JN = (JMAXN/NPER) + 1
DO 314 I=1,JN
NU = NU + 1
XOMU(NU) = XUPN(I)
YOMU(NU) = YUPN(I)
SMU(NU) = SMUN(I)
ETAMU(NU) = ETAUN(I)
314 CONTINUE
NCU = NCU-1
DO 315 I=2,NCU
NU = NU + 1
XOMU(NU) = XCUP(I)
YOMU(NU) = YCUP(I)
SMU(NU) = SSMUP
ETAMU(NU) = ETACU(I)

```

```

315 CONTINUE
  JO = (JMAXO/NPER) + 1
  DO 316 I=1,JO
    J = JC + 1 - I
    NU = NU + 1
    XOMU(NU) = XUPO(J)
    YOMU(NU) = YUPO(J)
    SMU(NU) = SMUO(J)
    ETAMU(NU) = ETAUO(J)
316 CONTINUE
  DO 317 I=2,11
    NU = NU + 1
    XOMU(NU) = XSO(I)
    YOMU(NU) = YSO(I)
    SMU(NU) = SSMOUT
    ETAMU(NU) = ETASO(I)
317 CONTINUE

  RETURN
  END
$IEFTC INPU2  DECK
  SUBROUTINE INPUT2 (XM,YM,VVCR,ETAM,NN)

  LOGICAL ERROR,TRANS,SEPRN

  COMMON/CODE/KCDE,DELSRL,DELSRU
  COMMON/LNK2/UPMAC,SMOUT,ALP1M,CSTAR,SIGMAO,GSTARO,GSTARI,GAMMA,
* CONVER,FECONV
  COMMON/CTOBL/ALPH1,SPA,XMAX,TF,NTURBU,NTURBL,CTHETU,CTHETL
  COMMON/C1/GAM,R,PTZ,TTZ,UPMACH,NST,NVP,NTURB,KPVM,KEM,KSMTH,
1KSPLN,KLE,KATCH,CTHET,DLAM,TLAM,DTURB,TTURB,KFRE,KGRAD,KSDE,KLAM,
2KMAIN,KPROF,X(100),Y(100),PPES(100),UF(100),ME(100),POPTZ(100),
3VOVCR(100),TWAL(100),ETA(100)
  COMMON/C3/XOM(100),YOM(100),S(100),SOL(100),AE(100),TSE(100),
1TAWL(100),TAWT(100),TBAR(100),RW(100),SW(100),SUTHL(100),
2RHSW(100),RHSE(100),HEADW(100),HEADE(100),NUW(100),MUBAR(100),
3AA(100),RB(100),FF(100),DUDS(100),DMDS(100),DMDL(100)
  COMMON/C9/ERROR,TRANS,SEPRN

  DIMENSION XM(NN),YM(NN),VVCR(NN),ETAM(NN)

C  THE V/VCR DISTRIBUTION IS RECEIVED FROM THE ROTCR PROGRAM
C  IT WILL BE USED TO CALCULATE THE PRESSURE DISTRIBUTION

  ERROR = .FALSE.
  TRANS = .FALSE.
  SEPRN = .FALSE.
C  BOUNDARY LAYER SETUP
  KATCH = 0
  KSPLN = 1
  KLE = 1
  TLAM = 0.
  DLAM = 0.
  TTURB = 0.
  DTURB = 0.

  NST = NN
  DO 2 I=1,NST
    XOM(I) = XM(I)

```



```

YOM(I) = YM(I)
X(I) = XOM(I)*XMAX
Y(I) = YOM(I)*XMAX
TWAL(I) = TTZ
ETA(I) = ETAM(I)
2 VOVCR(I) = VVCR(I)
ALPH1 = -ALF1M
SPA = XMAX/SIGMAO

IF (KCODE .NE. 0) GO TO 3
WRITE (6,100)
100 FORMAT (1H1,53X,23H*** UPPER SURFACE ***)
GO TO 4
3 WRITE (6,101)
101 FORMAT (1H1,53X,23H*** LOWER SURFACE ***)
4 WRITE (6,1020) NST,NVP,GAM,B,PTZ,TTZ,UPMACH,XMAX,NTURB,CTHET,TE
1020 FORMAT (1H0/5X,22HBOUNDARY LAYER - INPUT//5X,3HNST,5X,3HNVP,7X,
* 3HGAM,11X,1HR,12X,3HPTZ,11X,3HTTZ,9X,6HUPMACH,8X,4HXMAX,7X,
* 5HNTURB,6X,6HCTHETA,7X,2HTE/4X,2(I3,5X),F7.3,3(5X,F9.2),5X,F8.4,
* 5X,F8.5,6X,I3,7X,F7.3,5X,F7.5//)
WRITE (6,1090) KPRE,KGRAD,KSDF,KLAM,KMAIN,KPROF,KEM
1090 FORMAT (/6X,4HKPRE,7X,5HKGRAD,7X,4HKSDE,8X,4HKLAM,7X,5HKMAIN,7X,5HK
1PROF,7X,5HKEM /7X,I2,9X,I2,10X,I2,10X,I2,9X,I2,10X,I2,10X,I2//)
WRITE (6,1032) (XOM(I),YOM(I),VVCR(I),TWAL(I),I=1,NST)
1032 FORMAT (8X,3HXOM,7X,3HYOM,9X,4HVVCR,9X,4HTWAL/(5X,F8.5,2X,F8.5,2X,
1F11.5,3X,F9.2))

IF (NST.GT.100.OR.NTURB.GT.NST.OR.KFM.LT.0.OR.KEM.GT.1.OR.KSPLN.LT.
10.OR.KSPLN.GT.1.OR.KLE.LT.0.OR.KLE.GT.1.OP.KATCH.LT.0.OP.KATCH.GT.
21) GO TO 70
RETURN

70 ERROR = .TRUE.
WRITE (6,117C)
1170 FORMAT (///10X,48HERROR IN INPUT DATA.  PECHECK INPUT INSTRUCTIONS
1)
RETURN

END
$IEFTC BLAYFR DECK
SUBROUTINE ELAYR

C BOUNDARY LAYER MAIN PROGRAM

LOGICAL ERRCR,TRANS,SEPRN
REAL ME

COMMON/CODE/KCODE,DELSBL,DELSRU
COMMON/CTOBL/AL,SP,XMAX,TE,NTURBU,NTURBL,CHTETU,CTHETL
COMMON/C1/GAM,R,PTZ,TTZ,UPMACH,NST,NVP,NTURB,KPVM,KEM,KSMTH,
1KSPLN,KLF,KATCH,CTHET,DLAM,TLAM,DTURB,TTURB,KPRE,KGRAD,KSDE,KLAM,
2KMAIN,KPROF,X(100),Y(100),PRFS(100),UE(100),ME(100),POPTZ(100),
3VOVCR(100),TWAL(100),ETA(100)
COMMON/C2/PSZ,TSZ,UZ,ASZ,ATZ,RHSZ,RHTZ,MUSZ,MUTZ,MUSZ,NUTZ,CP,
1PR,TC,ARCL
COMMON/C3/XOM(100),YOM(100),S(100),SOL(100),AE(100),TSE(100),
1TAWL(100),TAWT(100),TRAP(100),RW(100),SW(100),SUTEL(100),
2RHSW(100),RHSF(100),HEADW(100),HEADE(100),NUW(100),MURAR(100),
3AA(100),BB(100),FF(100),DUDS(100),DMDS(100),DMDL(100)
COMMON/C4/THET(100),DELSR(100),DELTA(100),FCRM(100),

```

```

1FORMI(100),FCRMTR(100),PTH(100),RHTI(100),CF(100),
2TAUW(100),NUSS(100),DTDY(100),HTRAN(100),CFN(100)
COMMON/C9/ERROR,TRANS,SEPRN

```

```

CALL PRECAL
IF (ERROR) RETURN
CALL LAMNAP
IF (ERROR) RETURN
IF (SEPRN) GO TO 20
IF (.NOT. TRANS) GO TO 20
CALL TUBBLN
IF (ERROR) RETURN
20 CALL PROFIL
IF (KODE.EQ.1) GO TO 9
DELS = DELSR(NST)
THETS = THET(NST)
KODE = 1
RETURN
9 DELP = DELSR(NST)
THETP = THET(NST)
ALPH1 = AL
SPA = SP
CALL AFMIX (ALPH1,DELS,DELP,THETS,THETP,TE,SPA,ME(NST))
KODE = 0
RETURN

```

END

```

$IEFTC PRECL LIST,DECK
SUBROUTINE PRECAL

```

```

COMMON/GAMPM/GMP1,GMM1
COMMON/CTOBL/DUM(2),XMAX,DUM2(5)
COMMON/C1/GAM,B,PTZ,TTZ,UPMACH,NST,NVF,NTURB,KPVM,KEM,KSMTH,
1KSPIN,KLE,KATCH,CTHET,DLAM,TLAM,DTURB,TTURB,KPRE,KGRAD,KSDE,KLAM,
2KMAIN,KPROF,X(100),Y(100),PRES(100),UE(100),ME(100),POPTZ(100),
3VOVCR(100),TWAL(100),ETA(100)
COMMON/C2/PSZ,TSZ,UZ,ASZ,ATZ,RHSZ,RHTZ,MUSZ,MUTZ,NUSZ,NUTZ,CP,
1PR,TC,ARCL
COMMON/C3/XOM(100),YOM(100),S(100),SOL(100),AE(100),TSE(100),
1TAWL(100),TAWT(100),TBAR(100),RW(100),SW(100),SUTHL(100),
2RHSW(100),RHSE(100),HEADW(100),HEADE(100),NUW(100),MUBAR(100),
3AA(100),BB(100),FF(100),DUDS(100),DMDS(100),DMDL(100)
COMMON/C9/ERROR,TRANS,SEPRN
DIMENSION SDER(100),CMU(20),CPR(20),CTC(20)
REAL MUSZ,MUTZ,NUSZ,NUTZ,MUSLE,MUSLM,ME,NUW,MUBAR
LOGICAL ERROR,TRANS,SEPRN

```

```

C
C READ DATA FOR MU, PR, AND TC CURVE FITS
C

```

```

DATA(CMU(I),I=1,5)/-.01945170,1.3019531,-.34511323,
1.068277826,-.00566593/
DATA(CPR(I),I=1,5)/.8557,-.234136,.1078624,
1-.0236214,.00202863/
DATA(CTC(I),I=1,5)/-.03839323,1.2697427,-.30911252,
1.08743781,-.009674725/

```

```

C
GMP1 = GAM + 1.
GMM1 = GAM - 1.

```

C INITIALIZE STATIC AND TOTAL PARAMETERS

```

C
  TSLE= 518.688
  TSLM= 298.160
  MUSLE= 3.711402E-7
  MUSLM= 1.777029E-5
  TCSLE= 3.202206E-3
  TCSLM= 2.561796E-2
  TSZ = TTZ/(1.+GMM1/2.*UPMACH**2)
  PSZ = PTZ*(TSZ/TTZ)**(GAM/GMM1)
  RHSZ= PSZ/R/TSZ
  RHTZ= PTZ/R/TTZ
  ASZ= SQRT(GAM*R*TSZ)
  ATZ= SQRT(GAM*R*TTZ)
  UZ= UPMACH*ASZ
  CP = R*GAM/GMM1
  IF (KEM.FC.1) GO TO 10
  TCON= 198.60
  TR1= TSZ/TSLE
  TR2= TTZ/TSLE
  GO TO 20
10 TCON= 110.33
  TR1= TSZ/TSLM
  TR2= TTZ/TSLM
20 CALL CURVFT(CFR,PR,TR1,0,4,0)
  CALL CURVFT(CTC,TC,TR1,0,4,0)
  CALL CURVFT(CMU,MUSZ,TR1,0,4,0)
  CALL CURVFT(CMU,MUTZ,TR2,0,4,0)
  IF (KEM.EQ.1) GO TO 30
  TC= TC*TCSLE
  MUSZ= MUSZ*MUSLE
  MUTZ= MUTZ*MUSLE
  GO TO 40
30 TC= TC*TCSLM
  MUSZ= MUSZ*MUSLM
  MUTZ= MUTZ*MUSLM
40 MUSZ= MUSZ/RHSZ
  MUTZ= MUTZ/RHTZ

```

C
C CALCULATE GEOMETRY RATIOS AND ARC LENGTHS

```

C
  S(1)= 0.
  DO 50 I=2,NST
50 S(I)= S(I-1)+SQRT((X(I)-X(I-1))**2+(Y(I)-Y(I-1))**2)
  ARCL= S(NST)
  DO 60 I=1,NST
60 SOL(I)= S(I)/ARCL

```

C
C CALCULATE PREFS,UE,ME,POPTZ,AND VOVCR AT EACH STATION

```

C
  VELOCITY OVER CRITICAL VELOCITY GIVEN AS INPUT
150 DO 160 I=1,NST
  POPTZ(I) = (1.-GMM1/GMP1*VOVCR(I)**2)**(GAM/GMM1)
  UE(I) = SQRT(2.*GAM/GMM1*PTZ/RHTZ*(1.-POPTZ(I)**(GMM1/GAM)
  *))
  TSE(I) = TTZ-UE(I)**2/(2.*CP)
  AE(I) = SQRT(GAM*R*TSE(I))
  ME(I) = UE(I)/AE(I)
160 PRES(I) = PCPTZ(I)*PTZ

```

```

C
C PRINT INITIAL CALCULATED PARAMETERS
C
170 WRITE(6,1000)
    WRITE(6,1010) PSZ,TSZ,UZ,ASZ,ATZ,RHSZ,PHTZ,MUSZ,MUTZ,NUSZ,NUTZ,CP,
    1PR,TC,ARCL
    WRITE(6,1020) (I,PRES(I),UE(I),ME(I),POPTZ(I),VOVCR(I),I=1,NST)
C
C
C PRINT GEOMETRY PARAMETERS
C
200 IF (KPRE.NE.1) GO TO 210
    WRITE(6,1030) (I,X(I),Y(I),S(I),XCM(I),YCM(I),SOL(I),I=1,NST)
C
C CALCULATE OTHER NECCESSARY PARAMETERS AT EACH STATION
C
210 DO 220 I=1,NST
    TEM1 = 1.+GMM1/2.*MF(I)**2
    RHSW(I) = PRES(I)/R/TWAL(I)
    RHSE(I) = PRES(I)/R/TSE(I)
    HEADW(I) = .5*RHSW(I)*UE(I)**2
    HEADE(I) = .5*PHSE(I)*UF(I)**2
    SW(I) = TWAL(I)/TTZ-1.
    SUTHL(I) = SQRT(TWAL(I)/TTZ)*(TTZ+TCOV)/(TWAL(I)+TCOV)
    NUW(I) = SUTHL(I)*NUTZ*(1.+SW(I))**2*TEM1*(GAM/GMM1)
    RW(I) = UE(I)*S(I)/NUW(I)
    TAWL(I) = TSE(I)*(1.+PR**(1./2.)*(TEM1-1.))
    TAWT(I) = TSE(I)*(1.+PR**(1./3.)*(TEM1-1.))
    TBAP(I) = .5*(TWAL(I)+TSE(I))+.22*PR**(1./3.)*(TTZ-TSE(I))
    MUBAR(I) = MUTZ*SUTHL(I)*TBAP(I)/TTZ
    BB(I) = ME(I)*ATZ/NUTZ*(TSE(I)/TTZ)*(GMP1/(2.*GMM1))
    AA(I) = BB(I)*TSE(I)/TBAR(I)*(MUBAR(I)/MUTZ)**.268
    FF(I) = 1.+ .1599*ME(I)**2+.60*SW(I)+.2101*SW(I)*MF(I)**2+.0114*ME(I)
    1**4+.0180*SW(I)*ME(I)**4+.1825*SW(I)**2+.0735*SW(I)**2*ME(I)**2
    2+.0073*SW(I)**2*ME(I)**4
220 CONTINUE
C
C FINITE DIFFERENCE METHOD USED TO CALCULATE VELOCITY AND MACH NUMBER
C GRADIENTS ALONG THE SURFACE
C
    DUDS(1) = (UE(2) - UE(1))/(S(2) - S(1))
    DMDS(1) = (ME(2) - ME(1))/(S(2) - S(1))
    DO 230 I=2,NST
        IM = I - 1
        IF (I.EQ. NST) GO TO 230
        IP = I + 1
        DUDS(I) = (UE(IP)-UE(IM))/(S(IP)-S(IM))
        DMDS(I) = (ME(IP)-ME(IM))/(S(IP)-S(IM))
230 CONTINUE
    DUDS(NST) = (UE(NST) - UE(IM))/(S(NST) - S(IM))
    DMDS(NST) = (ME(NST) - ME(IM))/(S(NST) - S(IM))
240 DO 250 I=1,NST
250 DMDL(I) = ARCL*DMDS(I)
C
C PRINT OTHER CALCULATED PARAMETERS
C
    IF(KPRE.NE.1) GO TO 260
    WRITE(6,1050) (I,AE(I),ISE(I),TWAL(I),TAVL(I),TAWT(I),TEAR(I),
    1I=1,NST)

```

```

        WRITE(6,1060) (I,RW(I),SW(I),SUTHL(I),PHSW(I),PHSE(I),HEADW(I),
1HEADE(I),NUW(I),MUBAR(I),I=1,NST)
260 IF(KGPAD.NE.1) GO TO 270
        WRITE(6,1070)
        WRITE(6,1080) (I,DUDS(I),DMDS(I),DMDL(I),I=1,NST)
C
C CHECK FOR IMPROPER INPUT
C
270 DO 280 I=2,NST
    IF (UF(I).NE.0.) GO TO 280
    ERROR=.TRUE.
    WRITE(6,1090)
    RETURN
280 CONTINUE
    RETURN
C
C FORMAT STATEMENTS
C
1000 FORMAT(1H1///4X,24HPRELIMINARY CALCULATIONS///)
1010 FORMAT(5X,10HPSZ      = F12.5/5X,10HTSZ      = F10.4/5X,10HUZ      =
1 F11.5//5X,10HASZ      = F11.4/5X,10HATZ      = F11.4//5X,10HPSZ
2= G15.7/5X,10HRHTZ      = G15.7//5X,10HMUSZ      = G15.7/5X,10HMUTZ
3 = G15.7//5X,10HNUSZ      = G15.7/5X,10HNUTZ      = G15.7//5X,10HCP
4 = F11.4/5X,10HPR      = F9.5/5X,10HPC      = G15.7/5X,10HARCL
5 = F8.4//)
1020 FORMAT(/1X,7HSTATION,7X,4HPHFS,13X,2HUF,12X,2HME,11X,5HPOPTZ,9X,5H
1VCVCR/(2X,I3,5X,F12.5,3X,F12.5,4X,F10.6,4X,F10.6,4X,F10.6))
1030 FORMAT(///1X,7HSTATION,7X,1HX,12X,1HY,12X,1HS,12X,3HXOM,9X,3HYOM,
19X,3HSOL/(2X,I3,3X,F12.5,1X,F12.5,1X,F12.5,4X,F9.5,3X,F9.5,3X,F9.5
2))
1050 FORMAT(///1X,7HSTATION,5X,2HAF,10X,3HTSE,9X,4HTWAL,8X,4HTAWL,8X,4H
1TAWT,8X,4HTRAR/(2X,I3,4X,F9.3,5(4X,F8.3)))
1060 FORMAT(///1X,7HSTATION,11X,2HRW,6X,2HSW,4X,5HSUTHI,7X,4HPSW,12X,4
1HRHSE,8X,5HHEADW,4X,5HFACE,9X,3HNUW,12X,5HMURAR/(2X,I3,3X,F15.1,2
2X,F4.1,1X,F7.3,2X,G14.6,2X,G14.6,1X,F8.3,1X,F8.3,2X,G14.6,2X,G14.6
3))
1070 FORMAT(1H1///21X,17HSURFACE GRADIENTS///)
1080 FORMAT(1X,7HSTATION,13X,4HDUDS,15X,4HDMDS,15X,4HDMDL/(2X,I3,4X,F18
1.6,1X,F18.6,1X,F18.6))
1090 FORMAT(/////10X,83HTHERE IS A STAGNATION POINT AT A STATION OTHER
1THAN STATION 1. THIS IS NOT ALLOWED)
    END
$IFFTC LAMNA LIST,DECK
    SUBROUTINE LAMNAR

    COMMON/GAMPE/GMP1,GMP1
    COMMON/CTOBL/DUM(2),XMAX,DUM2(5)
    COMMON/C1/GAM,R,PTZ,TTZ,UPMACH,NST,NVP,NTNPR,KPVM,KEM,KSMTH,
1KSPIN,KLF,KATCH,CTHET,DIAM,TLAM,DTURP,TTURP,KPRE,KGPAD,KSDF,KLAM,
2KMAIN,KPROF,X(100),Y(100),PPFS(100),UF(100),ME(100),POPTZ(100),
3VOVCR(100),TWAL(100),ETA(100)
    COMMON/C2/PSZ,TSZ,UZ,ASZ,ATZ,PHSZ,PUTZ,PHSZ,MUTZ,NUSZ,NUTZ,CP,
1PR,TC,ARCL
    COMMON/C3/XOM(100),YOM(100),S(100),SOL(100),AE(100),TSE(100),
1TAWL(100),TAWT(100),TEAR(100),RW(100),SW(100),SUTHL(100),
2RHSW(100),RHSE(100),HEADW(100),HEADE(100),NUW(100),MUBAR(100),
3AA(100),BB(100),FF(100),DUDS(100),DMDS(100),DMDL(100)
    COMMON/C4/THET(100),DELSR(100),DELTA(100),FCPR*(100),
1FORMI(100),FCPMTR(100),PTH(100),RTHI(100),CF(100),

```

```

1TAUW(100),NUSS(100),DTDY(100),HTRAN(100),CRN(100)
COMMON/C5/SHAPL(100),SHAFL(100),B,NS
COMMON/C6/FTRAN,FORMS
COMMON/C7/INST,ITRAN,ISEP
COMMON/C9/ERROR,TRANS,SEPRN

```

```

DIMENSION CCRLN(100),CORML(100),SHFAR(100),DTH(100)
DIMENSION CCN(20),CRCP(20),CDIF(20),CSHR(20),CCRN(20),CDTH(20)
DIMENSION STAB(505),CTAP1(505),CTAP2(505)
REAL MUSZ,NUSZ,MUTZ,NUTZ,ME,NUW,MUEAF,NUSS,NUPW,KBAR,INT1,INT2
LOGICAL ERRCR,TRANS,SEPRN
EXTERNAL FUNCT,INT1,INT2

```

```

C
C READ DATA FOR CORLN(1), BCRIT, DIFF, SHEAF, CRN, AND DTH CURVE FITS
C

```

```

DATA(CCN(I),I=1,6)/-.08178,-.06670,-.03143,
1.00873,-.01657,-.01052/
DATA(CRCP(I),I=1,6)/5.47073,43.6053,227.198,
1-2067.04,-27172.7,13691.2/
DATA(CDIF(I),I=1,6)/903.785,26365.0,3.85695E+5,
11.11044E+6,-4.53853E+7,-7.70276E+7/
DATA(CSHR(I),I=1,16)/.224488,-1.91539,-9.894,-68.13488,
1-.001512,-1.4768,-10.52925,-152.2781,-.002406,-.015629,
1-1.45743,-126.23395,-.000752,-.005385,-.917838,-39.40644/
DATA(CCRN(I),I=1,16)/2.02056,-19.7211,-24.0495,-1400.002,
1-.050979,-10.88012,62.4419,-5081.76,-.014343,2.279845,
1129.7008,-6257.848,-.0270567,-1.677051,57.4397,-2552.266/
DATA(CDTH(I),I=1,16)/8.02829,-4.30978,88.8244,36.4336,
12.71101,-7.42259,242.293,-16.293,-.16394,-7.61942,286.9795,
164.11186,-.16758,-3.70289,130.8107,111.3276/

```

```

C
C INITIALIZE PARAMETERS
C

```

```

INST = 0
ITRAN = 0
ISEP = 0
CF(1) = 0.
TAUW(1) = 0.
NUSS(1) = 0.
DTDY(1) = 0.
HTRAN(1) = 0.
CRN(1) = 0.
RTRAN = 0.
IF (CTHET .GT. 0.) KATCH = 1

```

```

C
C CHECK CONSISTENCY OF INITIAL VALUES
C

```

```

IF (DIAM.GE.0..AND.TLAM.GE.0..AND.FTURB.GE.0..AND.TTURB.GE.0.)
1GO TO 10
ERROR = .TRUE.
WRITE(6,1000)
RETURN
10 IF (NTURB.NE.1) GO TO 30
ITRAN = 1
IF (DTURB.GT.0..AND.TTURB.GT.0.) GO TO 20
ERROR = .TRUE.
WRITE(6,1010)
RETURN

```

```

20 IF (UE(1).GT.0.) GO TO 240
   ERROR = .TRUE.
   WRITE(6,1020)
   RETURN
C
C BEGIN CALCULATION IN LAMINAR REGION - CHECK FOR INITIAL VALUES
C CALCULATE INITIAL CORRELATION NUMBER
C
30 IF (DLAM.EQ.0..AND.TLAM.EQ.0.) GO TO 70
   IF (UE(1).GT.0.) GO TO 40
   ERROR = .TRUE.
   WRITE(6,1030)
   RETURN
40 IF (TLAM.EQ.0.) GO TO 50
C INITIAL MOMENTUM THICKNESS WAS GIVEN
   TEM1 = 1.+GMM1/2.*ME(1)**2
   CORML(1) = -ATZ*TLAM**2/NUZT/SUTHL(1)/ARCL/TEM1**((3.-GAM)/
1(2.*GMM1))
   CORLN(1) = CORML(1)*DMDL(1)
   GO TO 90
C INITIAL DISPLACEMENT THICKNESS WAS GIVEN
50 IF (ABS(DMDL(1)).GE..0001) GO TO 60
   CORLN(1) = 0.
   TEM1 = 1.+GMM1/2.*ME(1)**2
   FORM(1) = 2.38411*(1.+(2.79-1.78*PR**5)*((1.+SW(1))*TEM1-1.))+ (4.6
15*PR**((1./3.)-3.65*PR*.5)*PR**5*(TEM1-1.))
   THET(1) = DLAM/FORM(1)
   CORML(1) = -ATZ*THET(1)**2/NUZT/SUTHL(1)/ARCL/TEM1**((3.-GAM)/(2.*
1GMM1))
   GO TO 90
60 IF (DMDL(1).GT.0.) CALL ROOTB(-1.,0.,DLAM,FUNCT,.5E-5,CORLN(1),SL)
   IF (DMDL(1).LT.0.) CALL ROOTB(0.,.2,DLAM,FUNCT,.5E-5,CORLN(1),SL)
   CORML(1) = CORLN(1)/DMDL(1)
   GO TO 90
C
C NO INITIAL LAMINAR VALUES GIVEN
C CALCULATE INITIAL CORRELATION NUMBER
C
C SHARP LEADING EDGE
70 IF (KLE.NE.1.AND.ABS(DMDL(1)).GE..0001) GO TO 80
   CORLN(1) = 0.
   CORML(1) = 0.
   GO TO 90
C STAGNATION POINT
80 CALL CURVET(CCN,CORLN(1),SW(1),0,5,0)
   CORML(1) = CCORLN(1)/DMDL(1)
   IF (CORML(1).LT.0.) GO TO 90
   ERROR = .TRUE.
   WRITE(6,1040)
   RETURN
C
C SOLVE LAMINAR DIFFERENTIAL EQUATION
C CALCULATE CORRELATION NUMBERS ALONG THE SURFACE
C
90 TEM1 = 1.+GMM1/2.*ME(1)**2
   TEM2 = (3.*GAM-1.)/(2.*GMM1)
   DEL = 0.002*ARCL
   SS = -DEL
   NTAB = 1

```

```

CTAB1(1)= CORLN(1)
CTAB2(1)= CORML(1)
STAB(1)= 0.
100 SS= SS+DEL
SSDEL = SS+DEL
CALL LGRNGE(S,SW,NST,SS,ANS1)
CALL LGRNGE(S,ME,NST,SS,ANS2)
CALL LGRNGE(S,ME,NST,SSDEL,ANS3)
CALL LGRNGE(S,DMDL,NST,SSDEL,ANS4)
A1= 0.43631-0.00367*ANS1+0.00481*ANS1**2+0.00651*ANS1**3
A2= 5.43220+2.25400*ANS1-0.06672*ANS1**2-0.20637*ANS1**3
A3= 4.51903-10.49775*ANS1-12.71732*ANS1**2-2.95270*ANS1**3
A4= 19.01831+62.76597*ANS1+115.00986*ANS1**2+62.53113*ANS1**3
A= A1-A3*CTAB1(NTAB)**2-2.*A4*CTAB1(NTAB)**3
B= A2+2.*A3*CTAB1(NTAB)+3.*A4*CTAB1(NTAB)**2

C
C FOR SW = 0.0
IF (CTAB1(NTAB).GE.-.1) GO TO 101
A=.3953
B=4.739
101 K1 = 0
SOL1 = SS/ARCL
SOL2 = SSDEL/ARCL
TEM3 = SIMPS1(SOL1,SOL2,INT1,K1)
IF (TEM3.EQ.0..OR.K1.EQ.0) GO TO 110
ERROR= .TRUE.
WRITE(6,1050)
RETURN
110 IF (NTAB.GT.1) TEM4= ANS2**(-B)*TEM1**TEM2
TEM1 = 1.+GMM1/2.*ANS3**2
TEM5= ANS3**(-B)*TEM1**TEM2
TEM6= -A*TEM5*TEM3
IF (NTAB.EQ.1) TEM7=0.
IF (NTAB.GT.1) TEM7= TEM5/TEM4*CTAB2(NTAB)
NTAB= NTAB+1
CTAB2(NTAB)= TEM6+TEM7
CTAB1(NTAB)= CTAB2(NTAB)*ANS4
STAB(NTAB)= SSDEL

C
C WHEN SW IS NOT EQUAL TO 0.0 ,CURVE FIT RANGE ON CORLN
C IS FROM -.32 TO .16
C
IF (CTAB1(NTAB) .GT. .50) GO TO 120
IF (SS.LT.ARCL) GO TO 100
120 IF (KSDE.NE.1) GO TO 130
WRITE(6,1060)
WRITE(6,1070) (STAB(I),CTAB1(I),I=1,NTAB)

C
C CALCULATE LAMINAR BOUNDARY LAYER PARAMETERS AT EACH STATION
C
130 IF (KLAM.NE.1) GO TO 140
WRITE(6,1080)
140 I= 0
150 I= I+1
IF (I.EQ.NTURE) ITRAN=-1
IF (S(I).LE.STAB(NTAB)) GO TO 160
WRITE(6,1090)
160 IF (KLE.EQ.1 .AND. I.EQ.1) GO TO 151
CALL LGRNGE(STAB,CTAB1,NTAB,S(I),CORLN(I))
CALL LGRNGE(STAB,CTAB2,NTAB,S(I),CORML(I))

```



```

C OBTAIN SHEAR, CRN, AND DTH FROM CURVE FITS VS CORLN AND SW
151 CALL CURVFT(CSHR,SHEAR(I),CORLN(I),SW(I),3,3)
    CALL CURVFT(CCRN,CRN(I),CORLN(I),SW(I),3,3)
    CALL CURVFT(CDTH,DTH(I),CORLN(I),SW(I),3,3)
C
C FOR SW = 0.0
    IF(CORLN(I).GE.-.1) GO TO 161
    SHEAR(I) = -1.2222*CORLN(I)+.26
    CRN(I) = -58.824 *CORLN(I)-.6765
    DTH(I) = -22.222*CORLN(I)+7.1112
C CALCULATE OTHER LAMINAR BOUNDARY LAYER PARAMETERS
161 TEM1 = 1.+GMM1/2.*ME(I)**2
    THET(I) = SQRT(-CORM1(I)*NUTZ*SUTHL(I)*ABCL/ATZ*TEM1**((3.-GAM)/
    1(2.*GMM1)))
    FORM(I) = (-1.1138*CORLN(I)+2.38411)*(1.+(2.79-1.78*PR**.5)*((1.+
    1SW(I))*TEM1-1.))+(4.65*PP**(1./3.)-3.65*PP**.5)*PP**.5*(TEM1-1.)
    DELSR(I) = THET(I)*FORM(I)
    RTH(I) = UE(I)*THET(I)/NUW(I)
    FORM1(I) = (FORM(I)-SQRT(PR)*(TEM1-1.))/((1.+SW(I))*TEM1)
    FORMTR(I) = FCPMI(I)*(1.+SW(I))
    DELTA(I) = THET(I)*(DTH(I)+(TEM1-1.)*(FCPMTR(I)+1.))
    SHAPL(I) = DELTA(I)**2/NUW(I)*DUDS(I)
    IF (I.EQ.1) GO TO 180
    CFW = 2.*SHEAR(I)*SQRT(-SOL(I)/ME(I)/CORM1(I))
    CF(I) = CFW/SQRT(RW(I))
    TAUW(I) = CF(I)*HEADW(I)
    NURW = CFW*PR**.3/CRN(I)
    NUSS(I) = NURW*SQRT(RW(I))
    DTDY(I) = NUSS(I)*(TAWL(I)-TWAL(I))/S(I)
    HTRAN(I) = TC*DTDY(I)
    IF (TAUW(I).GT.0.) GO TO 180
    IF (KATCH.NE.0) GO TO 170
    ISEP= I
    SEPPN= .TRUE.
    RETURN
170 ITRAN= -2
    GO TO 270
180 IF (I.EQ.1.AND.UE(1).EQ.0.) GO TO 190
    SHAPK(I) = NUTZ*RTH(I)**2*SUTHL(I)**2*(1.+SW(I))**4/ATZ/ME(I)**2/
    1FF(I)/ABCL*EMDL(I)*TEM1**((1./GMM1))
    GO TO 200
190 SHAPK(I) = 0.07
200 RTHI(I) = RTH(I)*SUTHL(I)*(1.+SW(I))**2/FF(I)/SQRT(TEM1)
C
C CALCULATE RCRIT TO CHECK FOR INSTABILITY AND TRANSITION
C
    CALL CURVFT(CPCR,RCRIT,SHAPK(I),0.5,0)
    IF (SHAPK(I) .GT. .07) RCRIT = 9.3163
    RCRIT= EXP(RCRIT)
    IF(INST.NE.0) GO TO 210
C
C CHECK FOR INSTABILITY
C
    IF(RTHI(I).LT.RCRIT) GO TO 270
    RINS= RTHI(I)
    INST= I
    GO TO 270
C
C CHECK FOR TRANSITION

```

```

C
210 K1= 0
    NS= I
    TEM= SIMPS1(SOL(INST),SOL(I),INT2,K1)
    IF (TEM.EQ.0..OR.K1.EQ.0) GO TO 220
    ERROR= .TRUE.
    WRITE(6,1100)
    RETURN
220 KBAR= TEM/(SOL(I)-SOL(INST))
    CALL CURVFT(CDIF,DIFF,KPAR,0.5,0)
    IF (KBAR .GT. .03) DIFF = 44000.*KPAR+700.0
    RTRAN= RINS+DIFF
    IF(RTHI(I).LT.RTPAN) GO TO 270
    IF (I.LT.NTURE) GO TO 270
    ITRAN= -1
    GO TO 270
230 ITRAN= I

C
C COMPUTE INITIAL VALUES FOR TURBULENT SOLUTION
C
240 TRANS= .TRUE.
    IF (DTURB.EQ.0..AND.TTURB.EQ.0.) GO TO 260
    IF (DTURB.GT.0..AND.TTURB.GT.0.) GO TO 250
    ERROR = .TRUE.
    WRITE(6,1110)
    RETURN
250 THET(ITRAN)= TTURB
    FORM(ITRAN)= DTURB/TTURB
    TEM1 = 1.+GMM1/2.*ME(ITRAN)**2
    FORMI(ITRAN)= (FORM(ITRAN)-PR**(1./3.)*(TEM1-1.))/((1.+SW(ITRAN))
1*TEM1)
260 IF (CTHET.GT.0..AND.DTURB.EQ.0..AND.TTURB.EQ.0.) THET(ITRAN)=
1CTHET*THET(ITRAN)
    THETTR = THET(ITRAN)*(TSE(ITRAN)/TTZ)**(GMP1/(2.*GMM1))
    FTRAN= (ME(ITRAN)*ATZ*THETTR/NOTZ)**1.268
    IF (RTRAN .LE. 0.) GO TO 265
    FORMS= FORMI(ITRAN)-0.59389-0.06591*ALOG(PTRAN)+0.001272*(ALOG(PTE
1AN))**2
    IF (DTURB.GT.0..AND.TTURB.GT.0.) FORMS=FORMI(ITRAN)
    RETURN
265 FORMS = 1.4
    RETURN

C
C PRINT OUTPUT
C
270 IF (KIAM.NE.1) GO TO 280
    IF(INST.EQ.0 .OR. INST.EQ.I) WRITE(6,1120) I,CORLN(I),SHPAR(I),
1DTH(I),FORMTR(I),SHAPL(I),RTHI(I),SHAPK(I),PCRIT
    IF(INST.NE.0 .AND. INST.NE.I) WRITE(6,1130) I,CORLN(I),SHPAR(I),
1DTH(I),FORMTR(I),SHAPL(I),RTHI(I),KPAR,DIFF,FTRAN
    IF (ITRAN.EQ.-2) WRITE(6,1140)
280 IF(ITRAN.EQ.-1.OR.ITRAN.EQ.-2) GO TO 230
    IF (I.EQ.NST) RETURN
    GO TO 150

C
C FORMAT STATEMENTS
C
1000 FORMAT(//////,10X,60HA NEGATIVE INITIAL VALUE HAS BEEN GIVEN. THIS
1IS NOT ALLOWED)

```

```

1010 FORMAT(//////,10X,75HINITIAL VALUES WERE NOT GIVEN FOR THE TURBULEN
1T BOUNDARY LAYER AT STATION 1)
1020 FORMAT(//////,10X,80HINITIAL VALUES WERE GIVEN FOR THE TURBULENT BO
1UNDARY LAYER AT A STAGNATION POINT)
1030 FORMAT(//////,10X,94HINITIAL VALUES OTHER THAN ZERO WERE GIVEN FOR
1THE LAMINAR BOUNDARY LAYER AT A STAGNATION POINT)
1040 FORMAT(//////,10X,106HFOR THIS INPUT DATA STATION 1 IS ASSUMED TO B
1E A STAGNATION POINT, SINCE NO INITIAL THICKNESSES ARE GIVEN./
210X,118HIN THIS CASE PRESSURE SHOULD DECREASE INITIALLY. EITHER G
3IVE AN INITIAL VALUE FOR DISPLACEMENT OR MOMENTUM THICKNESS./
410X,60HOR BEGIN WITH A SHCPT REGION OF FAVORABLE PRESSURE GRADIENT
5.)
1050 FORMAT(//////,10X,37HERROR IN COMPUTING INTEGRAL FOR CORLN)
1060 FORMAT(1H1///7X,50HLAMINAR DIFFERENTIAL EQUATION - SOLUTION FOR CO
1RLN///5(24H      S      CCRLN      )//)
1070 FORMAT((5(F12.5,2X,F7.4,3X)))
1080 FORMAT(1H1///1X,59HLAMINAR CALCULATION OF INSTABILITY AND TRANSITI
1ON LOCATIONS///1X,7HSTATION,2X,5HCORLN,5X,5HSHEAR,5X,3HDLTH,6X,6HFO
2RMTR,4X,5HSHAPL,9X,4HRTHI,6X,5HSHAPK,9X,5HRCRIT,6X,4HKBAR,10X,4HDI
3FF,9X,5HTRAN)
1090 FORMAT(//////,10X,65HLAMINAR SOLUTION HAS PROCEEDED BEYOND THE RANG
1E WHERE IT IS VALID)
1100 FORMAT(//////,10X,36HERROR IN COMPUTING INTEGRAL FOR KBAR)
1110 FORMAT(//////,10X,64HIF INITIAL TURBULENT VALUES ARE GIVEN, THEY BO
1TH MUST BE NONZERO)
1120 FORMAT(I4,1X,5F10.4,1X,F12.1,1X,F10.5,1X,F12.1)
1130 FORMAT(I4,1X,5F10.4,1X,F12.1,24X,F12.5,1X,F12.1,1X,F12.1)
1140 FORMAT(//////,10X,85HLAMINAR SEPARATION HAS OCCURRED. ASSUMED TO BE
1 TRANSITION TO TURBULENT BOUNDARY LAYER)
      END
$IBFTC TURBL  LIST,DECK
      SUBROUTINE TURBLN

      COMMON/GAMP/GMP1,GMM1
      COMMON/C1/GAM,P,PTZ,TTZ,UPMACH,NST,NVF,NTURR,KPVM,KEM,KSMTH,
1KSPLN,KLE,KATCH,CTHET,DLAM,TLAM,DTUPR,TTUPR,KPPF,KGRAP,KSDE,KLAM,
2KMAIN,KPROF,X(100),Y(100),PRES(100),UF(100),ME(100),POPTZ(100),
3VOVCR(100),TWAL(100),ETA(100)
      COMMON/C2/PSZ,TSZ,UZ,ASZ,ATZ,RHSZ,PHTZ,MUSZ,MUTZ,NUSZ,NUTZ,CP,
1PR,TC,ARCL
      COMMON/C3/XCM(100),YOM(100),S(100),SOL(100),AE(100),TSE(100),
1TAWL(100),TAWT(100),TBAR(100),PW(100),SW(100),SUTHL(100),
2RHSW(100),RHSE(100),HFADW(100),HEADE(100),NUW(100),MUBAR(100),
3AA(100),BB(100),FF(100),DUDS(100),DMDS(100),DMDL(100)
      COMMON/C4/THET(100),DELSR(100),DELTA(100),FORM(100),
1FORMI(100),FCRMR(100),PTH(100),RTHI(100),CF(100),
1TAUW(100),NUSS(100),DTDY(100),HTRAN(100),CRN(100)
      COMMON/C6/FTPAN,FOPMS
      COMMON/C7/INST,ITPAN,ISEP
      COMMON/C8/XTAB(505),YTAB1(505),YTAB2(505),NTAB
      COMMON/C9/ERRCR,TRANS,SFPRN

      REAL MUSZ,NUSZ,MUTZ,NUTZ,ME,NUW,MUPAR,NUSS
      LOGICAL ERRCR,TRANS,SEPPN
C
C  SOLVE TURBULENT BOUNDARY LAYER DIFFERENTIAL EQUATIONS

```

```

C USING RUNGA-KUTTA
  CALL RUNKUT
  IF (KSDE.NE.1) GO TO 10
  WRITE (6,1000)
  WRITE (6,1010) (XTAB(I),YTAB1(I),YTAB2(I),I=1,NTAB)
10 DO 5 I=1,NTAB
  IF (YTAB2(I) .LE. 2.8) GO TO 5
  WRITE (6,100)
100 FORMAT (9H0 (TURBLN),5X,63HINCOMERESSIBLE FORM FACTOR GREATER THAN
  *2.8 - CASE TERMINATED)
  STOP
  5 CONTINUE

C
C CALCULATE TURBULENT BOUNDARY LAYFF PAFAMETERS AT EACH STATION
C
  DO 30 I=ITRAN,NST
  IF (S(I).LE.XTAB(NTAB)) GO TO 20
  ISEP = I-1
  SEPRN = .TRUE.
  RETURN
20 TEM1 = 1.+GMM1/2.*ME(I)**2
  CALL LGRNGE(XTAB,YTAB1,NTAB,S(I),F)
  THETTR= NUTZ*F**.7886/ME(I)/ATZ
  THET(I) = THETTR*(TTZ/TSE(I))** (GMP1/(2.*GMM1))
  RTH(I) = UE(I)*THET(I)/NUW(I)
  CALL LGRNGF(XTAB,YTAB2,NTAB,S(I),FORMT(I))
  FORMTR(I) = FORMI(I)*(1.+SW(I))
  FORM(I) = FORMTR(I)*TEM1+PR**(1./3.)*(TFM1-1.)
  DELSR(I) = THET(I)*FORM(I)
  POWER = 2.0/(FORMI(I)-1.0)
  IF (FORMI(I).LT.1.02) POWER=100.
  DELTA(I) = (1.+POWER)*DFLSR(I)
  CF(I) = 0.246*EXP(-1.561*FORMI(I))*(UE(I)*THET(I)/NUTZ/(TFM1**(1./
1GAM-1.)))*(-.268)*TSE(I)/TBAR(I)*(MUBAR(I)/MUTZ)**(.268)
  TAUW(I) = CF(I)*HEADE(I)
  IF (I.EQ.1) GO TO 30
  HTRAN(I) = CF(I)/2./PR**(2./3.)*PHSE(I)*UE(I)*CP*(TAWT(I)-TVAL(I))
  DTDY(I) = HTRAN(I)/TC
  NUSS(I) = S(I)*DTDY(I)/(TAWT(I)-TVAL(I))
  CRN(I) = CF(I)*RW(I)/NUSS(I)
30 CONTINUE
  RETURN
1000 FORMAT(1H1///5X,62HTURBULENT DIFFERENTIAL EQUATIONS - SOLUTION FOR
1 F AND FORMI///4(31H S F FORMI )//)
1010 FORMAT((4(F10.5,2X,F8.1,2X,F7.4,2X)))
  END
$IPFTC PROFIL LIST,DECK
  SUBROUTINE PROFIL

  COMMON/CCODE/KCODE,DELSRL,DELSRU
  COMMON/CTOBL/DUM(2),XMAX,DUM2(5)
  COMMON/LNK2/DUM3(3),CSTAR,DUM35(2),GSTAPI,DUM4(3)
  COMMON/C1/GAM,P,PTZ,TTZ,UPMACH,NST,NVP,NTURB,KPVM,KFM,KSMTH,
1KSPIN,KLE,KATCH,CTHET,DLAM,TLAM,DTURB,TTUPP,KPPE,KGRAD,KSDE,KIAM,
2KMAIN,KPROF,X(100),Y(100),PRES(100),UE(100),ME(100),POPTZ(100),
3VOVCR(100),TVAL(100),ETA(100)
  COMMON/C3/XOM(100),YOM(100),S(100),SOL(100),AF(100),TSE(100),
1TAWL(100),TAWT(100),TRAP(100),RW(100),SW(100),SUTHL(100),
2RHSW(100),RHSE(100),HEADW(100),HEADE(100),NUW(100),MUBAR(100),

```

```

3AA(100),BB(100),FF(100),DUDS(100),DMDS(100),DMDL(100)
COMMON/C4/THET(100),DELSR(100),DELTA(100),FORM(100),
1FORMI(100),FORMTR(100),RTH(100),RTHI(100),CF(100),
1TAUW(100),NUSS(100),DTDY(100),HTRAN(100),CRN(100)
COMMON/C5/SHAPL(100),SHAPK(100),B,NS
COMMON/C7/INST,ITRAN,ISEP
REAL ME,NUSS

```

C
C PRINT LOCATIONS OF INSTABILITY, TRANSITION, AND SEPARATION
C

```

IF (KMAIN.NE.1) GO TO 60
WRITE(6,1000)
IF (INST.EQ.0) GO TO 10
WRITE(6,1010) INST
GO TO 20
10 WRITE(6,1020)
20 IF (ITRAN.LE.1) GO TO 30
WRITE(6,1030) ITRAN
GO TO 40
30 WRITE(6,1040)
40 IF (ISEP.EQ.0) GO TO 50
WRITE(6,1050) ISEP
GO TO 60
50 WRITE(6,1060)

```

C
C PRINT LOCATIONS OF LAMINAR AND TURBULENT BOUNDARY LAYERS
C

```

60 IEND = ITRAN-1
IF (IEND.EQ.-1.OR.IEND.EQ.0) IEND=ISEP
IF (IEND.EQ.0) IEND=NST
IF (KMAIN.NE.1) GO TO 70
IF (ITRAN.EQ.1) WRITE(6,1070)
IF (ITRAN.NE.1) WRITE(6,1080) IEND
IF (ITRAN.EQ.0) WRITE(6,1090)
IF (ITRAN.EQ.1) WRITE(6,1100) ITRAN,IEND
70 IF (ITRAN.LE.1) GO TO 80
IEND = ISEP
IF (IEND.EQ.0) IEND=NST
IF (KMAIN.NE.1) GO TO 90
WRITE(6,1100) ITRAN,IEND

```

```

80 IF (KODE .EQ. 0) DELSRU = DELSR(IEND)/XMAX
IF (KODE .EQ. 1) DELSRL = DELSR(IEND)/XMAX

```

```

WRITE(6,2000)
2000 FORMAT (1H1,35X,29HDIMENSIONED BLADE COORDINATES,36X,24HANGLES OF
*ROTATION (DEG)//37X,11HUNCORRECTED,14X,9HCORRECTED,15X,10HTRANSLAT
$ED//1X,7HSTATION,12X,1HX,23X,1HY,21X,5HYCORR,18X,5HYTRAN)
SIGN = +1.
IF (KODE .EQ..0) SIGN = -1.
GSTAR = GSTAR*XMAX/CSTAR
DO 170 I=1,IEND
YCORR = Y(I) + SIGN*ABS(DELSR(I)/COS(ETA(I)))
YTRAN = YCORR - GSTAR
ETAD = 57.2957796*ETA(I)
WRITE(6,2001) I,X(I),Y(I),YCORR,YTRAN,ETAD
2001 FORMAT (2X,I3,9X,4(F9.5,15X),F9.4)
170 CONTINUE

```

```

C
C PRINT CALCULATED BOUNDARY LAYER PARAMETERS
C
      IF (KMAIN.NE. 1) GO TO 90
      WRITE(6,1110)
      WRITE(6,1120) (I,X(I),S(I),DFLSR(I),THET(I),DELTA(I),FORM(I),
1FORMI(I),I=1,IEND)
      WRITE(6,1130)
      WRITE(6,1140) (I,CF(I),TAUW(I),RTH(I),DTDY(I),NUSS(I),HTRAN(I),
1CRN(I),I=1,IEND)
C
C COMPUTE BOUNDS ON VELOCITY PROFILES
C
      90 IF (KPROF.NE.1) RETURN
      WRITE(6,1150)
      IF(ITRAN.NE.0) GO TO 100
      IL1= 2
      IL2= IEND
      IT1= 0
      IT2= 0
      GO TO 110
100 IL1= 2
      IL2= ITRAN-1
      IT1= ITRAN
      IT2= IEND
      IF (IT1.EQ.1) IT1=2
C
C CALCULATE AND PRINT LAMINAR BOUNDARY LAYER VELOCITY PROFILES
C
      110 NVP1= NVP+1
      IF (IL2.LT.IL1) GO TO 140
      DO 130 I=IL1,IL2
      WRITE(6,1160) I
      AAA= 2.+SHAPL(I)/6.
      BBB= -.5*SHAPL(I)
      CCC= -2.+.5*SHAPL(I)
      DDD= 1.-SHAPL(I)/6.
      DEL= DELTA(I)/FLOAT(NVP)
      YP= -DEL
      DO 120 J=1,NVP1
      YP= YP+DEL
      ETAA = YP/DELTA(I)
      YXMAX= YP/X(NST)
      UUE = ((DDD*ETAA+CCC)*ETAA+BBB)*ETAA+AAA)*ETAA
      U= UUE*UE(I)
120 WRITE (6,1180) ETAA,YP,YXMAX,U,UUE
130 CONTINUE
C
C CALCULATE AND PRINT TURBULENT BOUNDARY LAYER VELOCITY PROFILES
C
      140 IF(IT1.EQ.0) RETURN
      DO 160 I=IT1,IT2
      POWER= DELTA(I)/DELSR(I)-1.
      WRITE(6,1170) I,POWER
      DEL= DELTA(I)/FLOAT(NVP)
      YP= -DEL
      DO 150 J=1,NVP1
      YP= YP+DEL
      ETAA = YP/DELTA(I)

```

```

YXMAX = YP/XMAX
UUE = ETAA**(1./POWER)
U= UUE*UE(I)
150 WRITE (6,1180) ETAA,YP,YXMAX,U,UUE
160 CONTINUE
RETURN

```

C

C FORMAT STATEMENTS

C

```

1000 FORMAT(1H1///1X,36HPRINCIPAL BOUNDARY LAYER INFORMATION///)
1010 FORMAT (/10X,31HINSTABILITY OCCURS AT STATION ,I3)
1020 FORMAT (/10X,26HINSTABILITY DOES NOT OCCUR)
1030 FORMAT (/10X,30HTRANSITION OCCURS AT STATION ,I3)
1040 FORMAT (/10X,25HTRANSITION DOES NOT OCCUR)
1050 FORMAT (/10X,30HSEPARATION OCCURS AT STATION ,I3)
1060 FORMAT (/10X,25HSEPARATION DOES NOT OCCUR)
1070 FORMAT (/10X,37HLAMINAR BOUNDARY LAYER DOES NOT OCCUR)
1080 FORMAT (/10X,42HLAMINAR BOUNDARY LAYER - STATIONS 1 TO ,I3)
1090 FORMAT (/10X,39HTURBULENT BOUNDARY LAYER DOES NOT OCCUR///)
1100 FORMAT (/10X,35HTURBULENT BOUNDARY LAYER - STATIONS,2X,
113,6H TO ,I3///)
1110 FORMAT(/1X,7HSTATION,8X,1HX,12X,1HS,12X,5HDELSR,10X,4HTHET,11X,
15HDELTA,11X,4HFORM,10X,5HFORMI)
1120 FORMAT(2X,I3,3X,2F13.6,F14.6,1X,F14.6,1X,2F14.4)
1130 FORMAT(///1X,7HSTATION,6X,2HCF,13X,4HTAUW,11X,3HRTN,14X,4HDTDY,
113X,4HNUSS,10X,5HHTAN,12X,3HCRN)
1140 FORMAT(I5,F14.5,2X,F14.5,1X,F12.1,5X,F14.2,2X,F14.2,1X,
1F14.4,2X,F13.3)
1150 FORMAT(1H1///1X,17HVELOCITY PROFILES///)
1160 FORMAT (/1X,7HSTATION,1X,I5,2X,7HPROFILE/3X,7HY/DELTA,9X,
11HY,12X,6HY/XMAX,10X,1HU,12X,4HU/UE)
1170 FORMAT (/1X,7HSTATION,1X,I5,2X,7HPROFILE,28X,2HN=,1X,F6.2/3X,7HY/D
1ELTA,9X,1HY,12X,6HY/XMAX,10X,1HU,12X,4HU/UE)
1180 FORMAT(1X,F8.4,2X,2G15.6,2X,F9.2,6X,F8.4)
END

```

```

$IEFTC RUNKT LIST,DECK
SUBROUTINE RUNKUT

```

C

C RUNKUT SOLVES SIMULTANEOUS FIRST ORDER INITIAL VALUE

C ORDINARY DIFFERENTIAL EQUATIONS

C

```

COMMON/C1/GAM,R,PTZ,TTZ,UPMACH,NST,NVP,NTUPB,KPVM,KFM,KSMTH,
1KSPLN,KLE,KATCH,CTHET,DLAM,TLAM,DTUPB,TTURB,KPRE,KGRAD,KSDE,KLAM,
2KMAIN,KPROF,X(100),Y(100),PRES(100),UE(100),ME(100),POPTZ(100),
3VOVCR(100),TWAL(100),ETA(100)
COMMON/C3/XOM(100),YOM(100),S(100),SOL(100),AF(100),TSE(100),
1TAWL(100),TAWT(100),TBAR(100),BW(100),SW(100),SUTHL(100),
2RHSE(100),RHSE(100),HEADW(100),HEADE(100),NUW(100),MUBAR(100),
3AA(100),BB(100),FF(100),DUDS(100),DMDS(100),DMDL(100)
COMMON/C6/FTRAN,FORMS
COMMON/C7/INST,ITRAN,ISEP
COMMON/C8/XTAB(505),YTAB1(505),YTAB2(505),NTAB
DIMENSION YY(2),RY(2),YINC(2),DCT(2),RUK(2,4)
DOUBLE PRECISION XX,RX,YY,RY,RUK,DEL,DOT,
1TEM1,TEM2,TEM3,TEM4,TEM5,TEM6
REAL ME,NUW,MUBAR

```

C

C SET DEL SPACING AND STORE INITIAL VALUES

```

C      DEL= 0.002*S (NST)
10 YY (1) =FTRAN
   YY (2) = FORMS
   XX= S (ITRAN)
   NV=2
   NTAB = 1
   YTAB1 (1) = YY (1)
   YTAB2 (1) = YY (2)
   XTAB (1) = XX

C
C SOLVE FOR YY (1) AND YY (2) AT NEXT XX INCREMENT
C
C SAVE PREVIOUS YY (1) AND YY (2)
20 DO 30 J=1,NV
30 RY (J) = YY (J)
   RX= XX

C
C CALCULATE NEW YY (1) AND YY (2)
C
   DO 90 L=1,4
C PUT DIFFERENTIAL EQUATIONS IN THE FORM OF
C FIRST DERIVATIVE = REMAINDER OF EQUATION
   CALL LGRNGE (S,ME,NST,XX,ANS1)
   CALL LGRNGE (S,SW,NST,XX,ANS2)
   CALL LGRNGE (S,AA,NST,XX,ANS3)
   CALL LGRNGE (S,BB,NST,XX,ANS4)
   CALL LGRNGE (S,DMDS,NST,XX,ANS5)
   CALL LGRNGE (S,TBAR,NST,XX,ANS6)
   TEM1= 1.+(1.+ANS2)*YY (2)
   TEM2= .123*EXP (-1.561*YY (2)) *ANS3
   DOT (1) = 1.268*(-YY (1)/ANS1*ANS5*TEM1+TEM2)
   TEM3= YY (2) *(YY (2)+1.) **2*(YY (2)-1.)
   TEM4= 1.+ANS2*(YY (2)*YY (2)+4.*YY (2)-1.)/( (YY (2)+1.)*(YY (2)+3.))
   TEM5= (YY (2)*YY (2)-1.)*YY (2)/YY (1)* (.123*EXP (-1.561*YY (2)) *ANS3)
   TEM6= (YY (2)*YY (2)-1.)/YY (1)** (.7886)* (.011*(YY (2)+1.)*(YY (2)-1.)
1**2/YY (2)**2*TTZ/ANS6)*ANS4
   DOT (2) = -ANS5*.5/ANS1*TEM3*TEM4+TEM5-TEM6
C APPLY THE RUNGA-KUTTA SCHEMF
   DO 40 J=1,NV
40 RUK (J,L) = DEL*DOT (J)
   GO TO (50,50,70,90), L
50 DO 60 J=1,NV
60 YY (J) = RY (J)+RUK (J,L)/2.
   XX= RX+DEL/2.
   GO TO 90
70 DO 80 J=1,NV
80 YY (J) = RY (J)+RUK (J,L)
   XX= RX+DEL
90 CONTINUE
C INCREMENT THE DEPENDENT VARIABLES TO OBTAIN NEW YY (1) AND YY (2)
   DO 100 J=1,NV
   YINC (J) = (RUK (J,1)+2.*RUK (J,2)+2.*RUK (J,3)+RUK (J,4))/6.
100 YY (J) = RY (J)+YINC (J)
C
C STORE NEW COMPUTED VALUES IN A TABLE

```



```

C
NTAB = NTAB+1
YTAB1(NTAB) = YY(1)
YTAB2(NTAB) = YY(2)
XTAB(NTAB) = XX
IF (YTAB2(NTAB) .GT. 2.8) RETURN
IF (XX.IT.S(NST)) GO TO 20
RETURN
END
$IBFTC SPLIN DECK
SUBROUTINE SPLINE(X,Y,N,DYDX,D2YDX2)
C
C SPLINE FITS A SPLINE CURVE TO X AND Y
C AND CALCULATES FIRST AND SECOND DERIVATIVES AT THE SPLINE POINTS
C END POINT SECOND DERIVATIVES EQUAL THOSE AT ADJACENT POINTS
C
DIMENSION X(N),Y(N),DYDX(N),D2YDX2(N)
DIMENSION G(100),H(100)
G(1) = -1.
H(1) = 0.
N1 = N-1
IF (N1.LT.2) GO TO 20
DO 10 I=2,N1
A = (X(I)-X(I-1))/6.
B = (X(I+1)-X(I))/6.
C = 2.*(A+B)-A*G(I-1)
D = (Y(I+1)-Y(I))/(X(I+1)-X(I)) - (Y(I)-Y(I-1))/(X(I)-X(I-1))
G(I) = B/C
10 H(I) = (D-A*H(I-1))/C
20 D2YDX2(N) = H(N1)/(1.+G(N1))
DO 30 I=2,N
K = N+1-I
30 D2YDX2(K) = H(K)-G(K)*D2YDX2(K+1)
DYDX(1) = (X(1)-X(2))/6.*(2.*D2YDX2(1)+D2YDX2(2)) + (Y(2)-Y(1))/(X(2)-X(1))
DO 40 I=2,N
40 DYDX(I) = (X(I)-X(I-1))/6.*(2.*D2YDX2(I)+D2YDX2(I-1)) + (Y(I)-Y(I-1))/(X(I)-X(I-1))
RETURN
END
$IBFTC LGRNGV DECK
SUBROUTINE LGRNGE(X,Y,N,ARG,ANS)
C
C LINEAR INTERPOLATION (REPLACES 4-POINT LAGRANGE)
C
DIMENSION X(N),Y(N)
IF (ARG .GE. 0.80*X(1)) GO TO 2
WRITE (6,100) ARG,X(1),X(N)
100 FORMAT (9H0 (LGRNGE),5X,10HABSCISSA =,E13.5,27H IS OUT OF RANGE -
* X(1) =,E13.5,5X,6HX(N) =,E13.5/15X,27HEXTRAPOLATED VALUE RETURNED
*)
2 NM = N - 1
DO 3 I=2,NM
IF (ARG .GT. X(I)) GO TO 3
M = I
GO TO 4

```

```

3 CONTINUE
  IF (ARG .LE. 1.20*X(N)) GO TO 5
  WRITE (6,100) ARG,X(1),X(N)
5 M = N
4 ANS = Y(M) + (Y(M) - Y(M-1))/(X(M) - X(M-1))*(ARG - X(M))
  RETURN

  END
$IEFTC SIMP      DECK
  FUNCTION SIMPS1(X1,X2,FUNC,KSIG)
  DIMENSION V(200),H(200),A(200),B(200),C(200),P(200),E(200)
  LOGICAL SPILL
  DOUBLE PRECISION ANS,Q
  DATA TWO,THREE,FOUR,THIRTY/2.0,3.0,4.0,30.0/
  DATA T,NMAX,NSIG/3.0E-5,200,1/
C  INITIALIZE FIRST ELEMENTS OF ARRAYS.
  V=X1
  H=(X2-V)/TWO
  A=FUNC(V)
  B=FUNC(V+H)
  C=FUNC(X2)
  P=H*(A+FOUR*B+C)
  E=P
  ANS=P
  N=1
  FRAC=T
  SPILL=.FALSE.
10 TEST=ABS(FRAC*ANS)
  K=N
  DO 30 I=1,K
C  TEST MAGNITUDE OF 4TH ORDER ERROR IN THIS INTERVAL.
  IF (ABS(E(I)).LF.TEST) GO TO 30
  IF (N.LT.NMAX) GO TO 20
C  GO TO FINISH IF STORAGE IS FILLED UP.
  SPILL=.TRUE.
  KSIG=KSIG+NSIG
  GO TO 40
C  SUBDIVIDE INTERVAL AGAIN TO REDUCE 4TH ORDER ERROR.
20 N=N+1
  V(N)=V(I)+H(I)
  H(N)=H(I)/TWO
  A(N)=E(I)
  B(N)=FUNC(V(N)+H(N))
  C(N)=C(I)
  P(N)=H(N)*(A(N)+FOUR*B(N)+C(N))
  H(I)=H(N)
  B(I)=FUNC(V(I)+H(I))
  C(I)=A(N)
  Q=P(I)
  P(I)=H(I)*(A(I)+FOUR*B(I)+C(I))
  Q=P(I)+P(N)-Q
  ANS=ANS+Q
  E(I)=Q
  E(N)=Q
30 CONTINUE
C  TEST ALL INTERVALS AGAIN IF ANY WERE SUBDIVIDED THE LAST TIME.
  IF (N.GT.K) GO TO 10
40 Q=0.0
  DO 50 I=1,N
50 Q=Q+E(I)

```

```

C   TIGHTEN ERROR LIMIT IF TOTAL ACCUMULATED ERROR TOO LARGE.
    IF (ABS(Q/T) .LE. ABS(ANS) .OR. SPILL) GO TO 60
    FRAC=FRAC/TWO
    GO TO 10
C   FINISH CALCULATION.
60  SIMPS1=(ANS+Q/THIRTY)/THREE
    RETURN
    END
$IEFTC CURVF  DECK
    SUBROUTINE CURVFT(COEF,ANS,X,Y,NX,NY)
C
C   EVALUATE THE POLYNOMIAL FUNCTION, ANS=F(X,Y), USING COEFFICIENTS, COEF
C
    DIMENSION COEF(20)
    NX1 = NX+1
    NY1 = NY+1
    ANS = COEF(1)
    IF (X.EQ..0.AND.Y.EQ..0) RETURN
    IF (Y.EQ..0) GO TO 10
    IF (X.EQ..0) GO TO 30
    GO TO 50
10  DO 20 I=2,NX1
20  ANS = ANS+COEF(I)*X**(I-1)
    RETURN
30  DO 40 I=2,NY1
    K = (I-1)*NX1+1
40  ANS = ANS+COEF(K)*Y**(I-1)
    RETURN
50  ANS = .0
    DO 60 I=1,NY1
    DO 60 J=1,NX1
    K = (I-1)*NX1+J
60  ANS = ANS+COEF(K)*Y**(I-1)*X**(J-1)
    RETURN
    END
$IEFTC AFTMIX  LIST,DECK
    SUBROUTINE AFMIX (ALPH1,DELS,DELP,THETS,THETP,TE,SP,XMPS1)

    DIMENSION XXX(100),YYY(100),SSS(100)

    COMMON/GAMPM/GMP1,GMM1
    COMMON/C1/GAM,R,PTZ,TTZ,UPMACH,NST,NVF,NTURB,KPVM,KEM,KSMTH,
1KSPLN,KLE,KATCH,CTHET,DLAM,TLAM,DTURB,TTURB,KPRE,KGRAD,KSDE,KLAM,
2KMAIN,KPROP,X(100),Y(100),PRES(100),UE(100),ME(100),POPTZ(100),
3VOVCR(100),TVAL(100),ETA(100)
    COMMON/C3/XOM(100),YOM(100),S(100),SOL(100),AE(100),TSE(100),
1TAWL(100),TAWT(100),TBAR(100),RW(100),SW(100),SUTHL(100),
2RHSW(100),RHSE(100),HEADW(100),HEADE(100),NUW(100),MUBAR(100),
3AA(100),BB(100),FF(100),DUDS(100),DMDS(100),DMDL(100)
    COMMON/C4/THET,DELSR(100),DELTA(100),FCPM(100),
1FORMI(100),FORMTR(100),RTH(100),RTHI(100),CF(100),
1TAUW(100),NUSS(100),DTDY(100),HTRAM(100),CRN(100)
    EQUIVALENCE (X,XXX),(Y,YYY),(S,SSS),(NST,N)
    REAL ME

    KODE = 0
    WRITE(6,5)
5   FORMAT(1H1,20X,22HAFTERMIXING PROPERTIES)
    XALPH1 = ALPH1
    ALPH1 = ALPH1*0.017453

```

```

2  VVCR1 = SQRT((GMP1/2.*XMFS1**2)/(1.+GMM1/2.*XMFS1**2))
   XX=SP*COS(ALPH1)
   DELSRT= (DELS+DELP)/XX
   THETA= (THETS+THETP)/XX
   DTE = TE/XX
   AFS1 = GMM1/GMP1*VVCR1**2
   A = 1.0-DELSRT-DTE-THETA
   A1= 1.0-DELSRT-DTE
   IF (A .LE. 0.0) GO TO 16
   IF (KODE .EQ. 0) WRITE (6,1)
1  FORMAT (1H0,10X,39HROTOR WITH NO BOUNDARY LAYER CORRECTION)
   C = ((1.-AFS1)*GMP1/(2.*GAM)+(COS(ALPH1))**2*
1A*VVCR1**2)/(COS(ALPH1)*A1*VVCR1)
   D = VVCR1*SIN(ALPH1)*A/A1
   VXVCR2 = GAM*C/GMP1-SQRT((GAM*C/GMP1)**2
1-1.+GMM1/GMP1*D**2)
   GO TO 4
3  KODE=2
   WRITE(6,10)
10  FORMAT(1H0,10X,19HSUPERSONIC SOLUTION)
   VXVCR2 = GAM*C/GMP1+SQRT((GAM*C/GMP1)**2
1-1.+GMM1/GMP1*D**2)
4  DENCH = GMM1/GMP1*(D**2+VXVCR2**2)
   IF (DENCH.GE. 1.0) RETURN
   DENSR2 = (1.-GMM1/GMP1*(D**2+VXVCR2**2))
1** (1./GMM1)
   DENSR1 = 1./((1.+GMM1/2.*XMFS1**2)** (1./GMM1))
   PR2 = DENSR2**GAM
   PT2PT0=(DENSR1*VVCR1*COS(ALPH1)*A1)/(DENSR2*VXVCR2)
   PTOP2 = 1.0/(PT2PT0*PR2)
   EBAR2 = ((PT2PT0)**(-GMM1/GAM)-1.)/(PTOP2**
1(GMM1/GAM)-1.)
   ETAN = 1.0-EBAR2
   VVCR2 = SQRT(D**2+VXVCR2**2)
   XM2 = SQRT(((2./GMP1)*VVCR2**2)/(1.-(GMM1/GMP1)
1*VVCR2**2))
   T2TT0 = 1.-GMM1/GMP1*VVCR2**2
   ALPH2 = ATAN(D/VXVCR2)
   XMX1 = XMFS1*COS(ALPH1)
   XMX2 = XM2*CCS(ALPH2)
   ALPH2 = ALPH2*57.2958
   WRITE(6,6) XMFS1,SP,TE,XM2,VVCR1,XMX1,XMX2
6  FORMAT(1H0,7HXMFS1 =,F6.4,2X,9HSPACING =,F7.6,2X,4HTE =,F7.5,2X,
15HXM2 =,F6.4,2X,8HV/VCR1 =,F6.3,2X,6HXX1 =,F6.3,2X,6HXX2 =,F6.3)
   WRITE (6,7) XALPH1,ALPH2,PT2PT0,PTOP2,T2TT0,VVCR2,EBAR2,ETAN
7  FORMAT(1H0,6HALPH1=,F7.3,2X,6HALPH2=,F7.3,2X,8HPT2/PT0=,F7.4,2X,
17HPT0/P2=,F9.3,2X,7HT2/TT0=,F7.4,2X,7HV/VCR2=,F6.3,2X,6HEBAR2=,
2F7.5,2X,6HETA-N=,F6.4)
   IF (KODE .EQ. 2 ) RETURN
   IF( KODE.EQ.1 ) GO TO 3
16 SP = SP + (DELS + DELP)/COS(ALPH1)
   WRITE(6,20)
20  FORMAT (1H0,10X,36HROTOR WITH BOUNDARY LAYER CORRECTION)
   KODE = 1
   GO TO 2
   END

```

```

$IEFTC FUNC      DECK
      SUBROUTINE FUNCT (XX,FX,DFX,INF)

      COMMON/GAMPM/GMP1,GMM1
      COMMON/C1/GAM,R,PTZ,TTZ,UPMACH,NST,NVF,NTURB,KPVM,KEM,KSMTH,
      1KSPLN,KLE,KATCH,CTHET,DLAM,TLAM,DTURB,TTURB,KPRE,KGRAD,KUDE,KLAM,
      2KMAIN,KPROF,X(100),Y(100),PRES(100),UE(100),ME(100),POPTZ(100),
      3VOVCR(100),TWAL(100),ETA(100)
      COMMON/C2/PSZ,TSZ,UZ,ASZ,ATZ,PHSZ,RHTZ,MUSZ,MUTZ,NUSZ,NUTZ,CP,
      1PR,TC,ARCL
      COMMON/C3/XOM(100),YOM(100),S(100),SOL(100),AE(100),TSE(100),
      1TAWL(100),TAWT(100),TBAP(100),RW(100),SW(100),SUTHL(100),
      2RHSW(100),RHSE(100),HEADW(100),HEADE(100),NUW(100),MUBAR(100),
      3AA(100),BB(100),FF(100),DUDS(100),DMDS(100),DMDL(100)

      REAL MUSZ,NUSZ,MUTZ,NUTZ,ME,NUW,MUBAR

      INF = 0
      B1 = 1.+GMM1/2.*ME(1)**2
      B2 = 1.+(2.79-1.78*PR**5)*((1.+SW(1))*B1-1.)
      B3 = -NUTZ*SUTHL(1)*ARCL/ATZ/DMDL(1)*P1**((3.-GAM)/(2.*GMM1))
      B4 = -1.1138*B2
      B5 = 2.38411*B2+(4.65*PR**5*(1./3.)-3.65*PR**5)*PR**5*(B1-1.)
      FX = (B3*XX)**.5*(B4*XX+B5)
      IF (XX.EQ.0.) GO TO 10
      DFX = .5*(B3*XX)**(-.5)*B3*(B4*XX+B5)+B4*(B3*XX)**.5
      RETURN
10  INF = 1
      DFX = 1.E10
      RETURN
      END
$IEFTC ROTB      DECK
      SUBROUTINE ROOTB (A,B,Y,FUNCT,TOLERY,X,DFX)
C
C  ROOT FINDS A ROOT FOR (FUNCT-Y) IN THE INTERVAL (A,B)
C
      X1= A
      X2= B
      CALL FUNCT(X1,FX1,DFX,INF)
10  DO 30 I=1,20
      X= (X1+X2)/2.
      CALL FUNCT(X,FX,DFX,INF)
      IF ((FX1-Y)*(FX-Y).GT.0.) GO TO 20
      X2= X
      GO TO 30
20  X1= X
      FX1= FX
30  CONTINUE
      IF (ABS(Y-FX).LT.TOLERY) RETURN
      WRITE(6,1000) A,B,Y
      STOP
1000 FORMAT(///4X,49HROOT HAS FAILED TO CONVERGE IN THE GIVEN INTERVAL
      1/4X,3HA =,G14.6,10X,3HB =,G14.6,10X,3HY =,G14.6)
      END

```

\$IEFTC INTG1 DECK

REAL FUNCTION INT1(XX)

COMMON/GAMPM/GMP1,GMM1

COMMON/C1/GAM,R,PTZ,TTZ,UPMACH,NST,NVP,NTURB,KPVM,KEM,KSMTH,
1KSPLN,KLE,KATCH,CTHET,DLAM,TLAM,DTURE,TTUPB,KPRE,KGRAD,KSDE,KLAM,
2KMAIN,KPROF,X(100),Y(100),PRES(100),UE(100),ME(100),POPTZ(100),
3VOVCR(100),TWAL(100),ETA(100)

COMMON/C3/XCM(100),YOM(100),S(100),SOL(100),AE(100),TSE(100),
1TAWL(100),TAWT(100),TBAR(100),PW(100),SW(100),SUTHL(100),
2RHSW(100),RHSE(100),HEADW(100),HEADE(100),NUW(100),MUBAR(100),
3AA(100),EB(100),FP(100),DHDS(100),DMDS(100),DMDL(100)
COMMON/C5/SHAPL(100),SHAPK(100),B,NS

REAL ME,NUW,MUBAR,INT1

CALL LGRNGE(SOL,ME,NST,XX,ANS)

INT1 = ANS**((B-1.)/(1.+GMM1/2.*ANS**2))**

1((3.*GAM-1.)/(2.*GMM1))

RETURN

END

\$IEFTC INTG2 DECK

REAL FUNCTION INT2(XX)

COMMON/C1/GAM,R,PTZ,TTZ,UPMACH,NST,NVP,NTURB,KPVM,KEM,KSMTH,
1KSPLN,KLE,KATCH,CTHET,DLAM,TLAM,DTURB,TTURB,KPRE,KGRAD,KSDE,KLAM,
2KMAIN,KPROF,X(100),Y(100),PRES(100),UE(100),ME(100),POPTZ(100),
3VOVCR(100),TWAL(100),ETA(100)

COMMON/C3/XCM(100),YOM(100),S(100),SOL(100),AE(100),TSE(100),
1TAWL(100),TAWT(100),TBAR(100),RW(100),SW(100),SUTHL(100),
2RHSW(100),RHSE(100),HEADW(100),HEADE(100),NUW(100),MUBAR(100),
3AA(100),EB(100),FP(100),DHDS(100),DMDS(100),DMDL(100)

COMMON/C5/SHAPL(100),SHAPK(100),B,NS

REAL ME,NUW,MUBAR,INT2

IF (NS.LT.4) GO TO 10

CALL LGRNGE(SOL,SHAPK,NS,XX,INT2)

RETURN

10 DO 20 J=2,NS

IF (SOL(J).LT.XX) GO TO 20

INT2= SHAPK(J-1) + (SHAPK(J)-SHAPK(J-1)) * (XX-SOL(J-1)) / (SOL(J)-SOL(J-1))

RETURN

20 CONTINUE

RETURN

END

Lewis Research Center,

National Aeronautics and Space Administration,

Cleveland, Ohio, September 8, 1971,

113-34.

APPENDIX A

PROGRAM CHANGES FOR A GAS OTHER THAN AIR

The program gas properties are set up for air. This program can be easily changed so that it applies to gases other than air. The changes are all made in subroutine PRECAL.

The coefficients read in by the DATA statements of CMU, CPR, and CTC arrays must be changed. The equations for the curve fits have the following form:

$$\frac{\mu}{\mu_{sl}} = a_1 + b_1 \left(\frac{T}{T_{sl}} \right) + c_1 \left(\frac{T}{T_{sl}} \right)^2 + d_1 \left(\frac{T}{T_{sl}} \right)^3 + e_1 \left(\frac{T}{T_{sl}} \right)^4$$

$$Pr = a_2 + b_2 \left(\frac{T}{T_{sl}} \right) + c_2 \left(\frac{T}{T_{sl}} \right)^2 + d_2 \left(\frac{T}{T_{sl}} \right)^3 + e_2 \left(\frac{T}{T_{sl}} \right)^4$$

$$\frac{k}{k_{sl}} = a_3 + b_3 \left(\frac{T}{T_{sl}} \right) + c_3 \left(\frac{T}{T_{sl}} \right)^2 + d_3 \left(\frac{T}{T_{sl}} \right)^3 + e_3 \left(\frac{T}{T_{sl}} \right)^4$$

If the number of coefficients changes from five in any case, this must be reflected both in the DATA statements and later in the calls on CURVFT where CMU, CPR, and CTC are used. If the properties are put in a different form, these cards must be removed.

The sea-level reference values in U. S. customary and SI units of temperature (TSLE, TSLM), viscosity (MUSLE, MUSLM), and thermal conductivity (TCSLE, TCSLM) must be changed.

The value of Sutherland's constant (TCON) and the computation of k_{su} (SUTHL (1)) will have to be changed. A temperature-viscosity law of the following form was used:

$$\frac{\mu}{\mu_o} = k_{su} \left(\frac{T}{T_o} \right)$$

Where k_{su} for air was used:

$$k_{su} = \left(\frac{T}{T_o} \right)^{1/2} \left(\frac{T_o + TCON}{T + TCON} \right)$$

Where TCON is Sutherland's constant for air.

APPENDIX B

ADDITIONAL OUTPUT

This output is obtained when KPRE, KGRAD, KSDE, KLAM, and KPROF are set equal to 1.

The output corresponding to KPRE are the geometric variables:

X	X-coordinate, m; ft
Y	Y-coordinate, m; ft
S	surface length, x, m; ft
XOM	ratio of X to C*
YOM	ratio of Y to C*
SOL	ratio of surface length to total arc length

The next part of the output gives the local speed of sound and several temperatures:

AE	local free-stream speed of sound, m/sec; ft/sec
TSE	static temperature, K; °R
TWAL	wall temperature, K; °R
TAWL	laminar recovery temperature, K; °R
TAWT	turbulent recovery temperature, K; °R
TBAR	reference temperature, K; °R

The final part of this output gives

RW	Reynolds number at wall, $RW = (UE)(S)/\mu W$
SW	temperature function at wall
SUTHL	value of the coefficient in Sutherland's viscosity temperature
RHSW	static density based on the wall temperature, kg/m^3 ; slug/ft^3
RHSE	static density based on free-stream temperature, kg/m^3 ; slug/ft^3
HEADW	velocity head based on the density at wall, N/m^2 ; lbf/ft^2
HEADE	velocity head based on the free-stream density, N/m^2 ; lbf/ft^2

NUW	kinematic viscosity at wall, m^2/sec ; ft^2/sec
MUBAR	dynamic viscosity based on reference temperature, $(\text{N})(\text{sec})/\text{m}^2$; $(\text{lbf})(\text{sec})/\text{ft}^2$

Output corresponding to KGRAD contains the three gradients of velocity and Mach number along the surface computed by finite difference methods:

DUDS	dU/dx , sec^{-1}
DMDS	dM/dx , m^{-1} ; ft^{-1}
DMDL	$dM/dSOL$

Output corresponding to KSDE contains the numerical solution of the laminar and turbulent differential boundary-layer equations. In the laminar case, the solution is the correlation number CORLN. In the turbulent case, the solution is the incompressible form factor FORMI, and a function F of the momentum thickness. These solutions are printed with respect to the surface length S.

Output corresponding to KLAM contains the variables used in the laminar subroutine to check for the position of instability and transition. The three variables, RTHI (increasing from station to station) and RCRIT and RTRAN (decreasing from station to station), are used in this analysis. When RTHI becomes larger than RCRIT, instability has occurred. When RTHI becomes larger than RTRAN, transition is assumed to occur. The variables listed are

CORLN	correlation number
SHEAR	shear parameter
DTH	ratio of transformed displacement thickness to transformed momentum thickness
FORMTR	transformed form factor
SHAPL	Pohlhausen shape factor based on boundary-layer thickness
RTHI	incompressible momentum-thickness Reynolds number
SHAPK	dimensionless shape factor based on momentum thickness
RCRIT	critical incompressible momentum-thickness Reynolds number
KBAR	mean shape factor based on momentum thickness
DIFF	difference between transition and instability momentum-thickness Reynolds numbers
RTRAN	incompressible momentum-thickness Reynolds number used in checking for transition point

Output corresponding to KPROF contains the velocity profiles at each station along the surface. The output listed is

Y/Delta	ratio of distance normal to surface in y-direction in boundary-layer profile to boundary-layer thickness
Y	distance normal to surface in y-direction in boundary-layer profile, m; ft
Y/XMAX	ratio of Y to XMAX
U	velocity within boundary layer, m/sec; ft/sec
U/UE	ratio of U of free-stream velocity

REFERENCES

1. Vanco, Michael R.; and Goldman, Louis J.: Computer Program for Design of Two-Dimensional Supersonic Nozzle with Sharp-Edged Throat. NASA TM X-1502, 1968.
2. Goldman, Louis J.; and Scullin, Vincent J.: Analytical Investigation of Supersonic Turbomachinery Blading. I-Computer Program for Blading Design. NASA TN D-4421, 1968.
3. Goldman, Louis J.; and Vanco, Michael R.: Computer Program for Design of Two-Dimensional Sharp-Edged-Throat Supersonic Nozzle with Boundary Layer Correction. NASA TM X-2343, 1971.
4. McNally, William D.: Fortran Program for Calculating Compressible Laminar and Turbulent Boundary Layers in Arbitrary Pressure Gradients. NASA TN D-5681, 1970.
5. Stewart, Warner L.: Analysis of Two-Dimensional Compressible-Flow Loss Characteristics Downstream of Turbomachine Blade Rows in Terms of Basic Boundary-Layer Characteristics. NACA TN 3515, 1955.
6. Cohen, Clarence B.; and Reshotko, Eli: The Compressible Laminar Boundary Layer with Heat Transfer and Arbitrary Pressure Gradient. NACA TR 1294, 1956.
7. Sasman, Philip K.; and Cresci, Robert J.: Compressible Turbulent Boundary Layer with Pressure Gradient and Heat Transfer. AIAA J., vol. 4, no. 1, Jan. 1966, pp. 19-25.
8. Goldman, Louis J.: Analytical Investigation of Blade Efficiency for Two-Dimensional Supersonic Turbine Rotor Blade Sections. NASA TM X-2095, 1970.



020B 01 C2 UL 01 711203 S00903DS 720401
DEPT OF THE AIR FORCE
AF WEAPONS LAB (AFSC)
TECH LIBRARY/WLOL/
ATTN: E LOU BOWMAN, CHIEF
KIRTLAND AFB NM 87117

POSTMASTER: If Undeliverable (Section 158
Postal Manual) Do Not Return

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

— NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546